Online schema upgrades using Loopback Replication

Nagaraju Inturi

nagaraju.inturi@hcl.com

HCL PRODUCTS & PLATFORMS

Agenda

- Use cases
- Prerequisites
- How it works
- Configuration requirements
- Demo

Use cases

- Minimal down time maintenance operations
  - Table reorg operations like changing fragmentation strategy for large tables, changing page size
  - Slow alters which require copying all rows to new table partition
  - Upgrade database schema while old applications are still active
  - Database codeset migration, example from en_us.819 to en_us.utf8
- Rolling schema (and applications) upgrade
  - Need changes to database schema ?
    - Then create new version of the database with updated schema, and migrate data to new version of the schema with near zero downtime to applications

How it works?

- Solution is based on Enterprise Replication
- Source and target tables can be in the same database or in two different databases.
- Only support primary->target replication topology.

Prerequisites

- Logging must be enabled for the tables
  - Do not work for RAW tables, and tables in non-logged databases
- Table must have primary key, Unique key or ERKEY.

Configuration requirements

- SQLHOSTS file
  - Require two group definitions. 1) Regular 2) Pseudo ER groups

    g_er_server group - - i=1

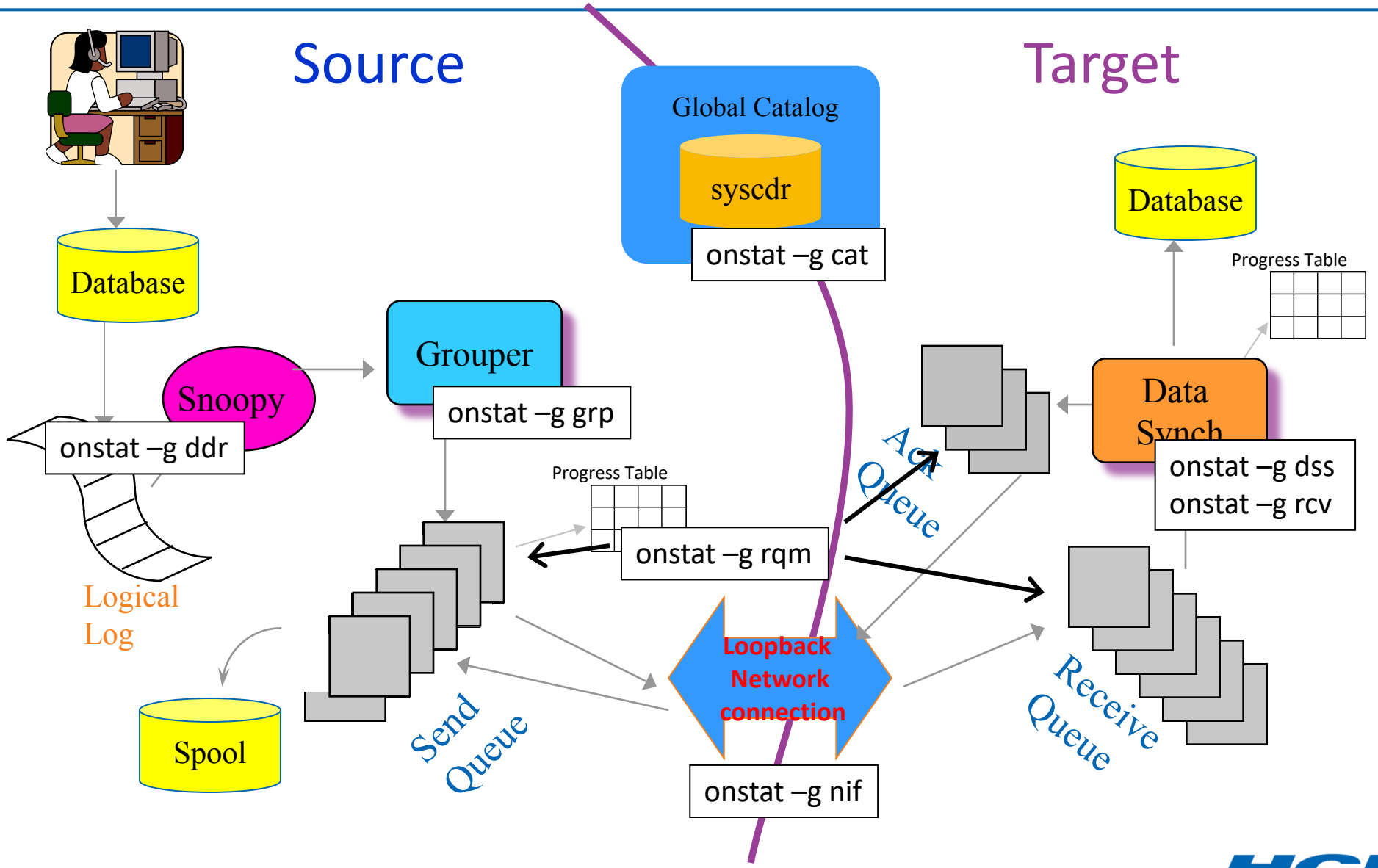
    *er_server onsoctcp *myhost 17001 g=g_er_server*

    g_loopback group - - i=2

    *loopback onsoctcp *myhost 17002 g= g_loopback*

- ONCONFIG
  - DBSERVERALIASES:
    - Server name associated with pseudo group need to be listed after server name associated with regular group
      - Example: *DBSERVERALIASES er_server,loopback*
    - Server name associated with pseudo group cannot be DBSERVERNAME

How it works?



Configuration requirements

- ONCONFIG
  - CDR_QUEUEMEM, CDR_QDATA_SBSPACE, CDR_DBSPACE, CDR_QHDR_DBSPACE and other ER config parameters needs to be set to appropriate values based on recommendations from Enterprise Replication guide.
  - CDR_SUPPRESS_ATSRISWARN 2,3,4   #to suppress warning ATS and RIS files

Define ER server

- Define ER for regular ER group
  - cdr define server --connect=g_er_server --init g_er_server
- Define ER for pseudo ER group sync(--sync option) with regular ER group
  - cdr define server --connect=g_loopback --nonroot --init --sync=g_er_server g_loopback
  - Note: Pseudo ER group needs to be defined as non-root ER server.

Define replicate definition

- Replicate needs to be defined using primary->target configuration
  - cdr define repl --connect=g_er_server test_t1 --conflict=always --scope=row --ats --ris --floatieee --master=g_er_server "P test@g_er_server:informix.t1" "select * from t1" "R test@g_loopback:informix.t2" "select * from t2"
  - cdr start repl --connect=g_er_server test_t1

Replication at Database level

- Template can be used to define replication across multiple tables from two different databases
  - cdr define template --connect=g_er_server ifxt_test_test --conflict=always --scope=row --ats --ris --floatieee --master=g_er_server --database=test --all
  - cdr realize template --connect=g_er_server ifxt_test_test "test@g_er_server"
  - cdr realize template --connect=g_loopback ifxt_test_test --target "test2@g_loopback"

ER commands supported with loopback replication

- Except grid commands, all other ER commands --including data synchronization(cdr sync) and repair(cdr check) commands -- works with loop back replication.
- Grid functionality isn’t applicable to loopback replication, and isn’t supported.
- Loopback replication can co-exist with existing Enterprise Replication nodes.

Sample procedure to reorg a table using loopback replication

- Create RAW table with appropriate fragmentation strategy
  - Create standard table if you have HDR/RSS/SDS secondary servers
- Copy data from old table to new table using external tables and pipes or using unload/load dbaccess statements.
- Create indexes
- Convert table type to “standard”.
- Define replication and repair data using ‘cdr sync repl’ or ‘cdr check --repair’ command.
- To rename table
  - Temporarily stop application(s)
  - Wait for queues to drain, command: onstat –g rqm brief
  - Delete replicate
- Use rename table DDL or synonym to point applications to new table.
- Restart applications

Sample procedure to migrate database codeset using loopback replication

- Create new database with new codeset and schema
- Unload data from old database using appropriate CLIENT and DB locales, and load data into new database use CLIENT_LOCALE set to unloaded data codeset format, and DB_LOCALE set to target database codeset.
- Define replication using –utf8 option and execute data re-synchronization task using ‘cdr check replset –repair’ or ‘cdr sync replset’ commands.
- If you plan to rename database
  - Stop old applications
  - Wait for queues to drain. Command: onstat –g rqm brief
  - Delete replicate definitions
- Use rename database DDL to point applications to new database.
- Start new applications

Demo