Oracle10g Streams: Data Replication Made Easy

Kirtikumar Deshpande

NYOUG
June 6, 2006
About Me

- Senior Oracle DBA
- Verizon Information Services
- Phone Directories Publication

verizon SUPERPAGES.com®

OakTable.net
Agenda

- What is Oracle Streams
- Components of Streams
- Data Replication Using Streams
- Turbulences in the Streams
- Streams Resources
- Q & A
What Is Oracle Streams?

- Information Sharing Solution
  - Within the same Database or Application
  - From one Database or Application to another
  - In Homogeneous or Heterogeneous Environment

- Data Replication
  - Offers flexible solution to Capture, Propagate and Apply data
What Can Streams Do?

- Data Replication
- Data Transformation
- Message Queuing
- Data Protection
- Data Warehouse Loading
Oracle Streams Components

3 Fundamental Components

- Capture
- Stage and Propagate
- Apply
Capture

- **Oracle Background process (c001 – c999)**
  - Invokes Logminer to read redo & archived log files
  - Captures Database Events
    - Database (Global) Level
    - Schema Level
    - Table Level
  - Applies defined Rules to events
  - Formats captured events into Logical Change Records (LCR)
    - Row/DML LCRs (SYS.LCR$ _ROW_RECORD)
    - DDL LCRs (SYS.LCR$ _DDL_RECORD)
Capture

- DML Changes not captured for columns
  - BFILE
  - ROWID
  - User Defined Types
  - Use Transparent Data Encryption

- Views showing unsupported tables
  - _DBA_STREAMS_UNSUPPORTED_10_2
  - _DBA_STREAMS_UNSUPPORTED_10_1
  - _DBA_STREAMS_UNSUPPORTED_9_2
Staging & Propagation

- LCRs are staged in a Queue in SGA
- Overflow written to a Disk Queue
- One Staging Area for all Events
- Self-describing data type SYS.ANYDATA
- Error Queue contains failed events
Staging & Propagation

- Events are dequeued:
  - For Consumption by local user application
  - To Propagate to Staging area of another Streams database
  - To Propagate to Apply process
- Events are purged after consumption by all recipients
- Propagation offers timing control
Apply

- **Oracle Background process (a001 – a999)**
  - Dequeues Events from Streams Queue
  - Applies Events to Database Objects
  - Performs various operations using an Apply Handler procedure
  - Detects data conflicts and applies resolution procedures

- **Define different Apply processes for**
  - User Defined Events
  - Captured Events
Rules And Rule Sets

- A Rule is a Database object
- Rules are used to limit which Events are:
  - Captured
  - Propagated
  - Applied
- Rules are grouped in Rule Sets
- Rule Sets are associated with Capture, Propagation and Apply process
Rules And Rule Sets

- Positive Rule Set
- Negative Rule Set
- System generated Rules
  - For DDL changes
  - For DML changes
  - Adequate and Sufficient in most cases
Rule-Based Transformations

- Any change to an event as a result of rule evaluating to True
- Transformation can be specified for events when:
  - Capturing
  - Propagating
  - Applying
Rule-Based Transformation

- To Change:
  - Data Format
  - Data Values (Mask Data)
  - Data Types
  - Column Names
  - Table Name
  - Owner Name
Directed Networks

Source DB
158.95.0.0

Intermediate DB
158.96.0.0

Destination DB
160.10.0.0

Destination DB
170.10.0.0
Data Replication Using Streams

1. Database Configuration
   • Archive Log Mode
   • Initialization Parameters
2. Create Streams Administrator Account
3. Apply (Target) Site Configuration
4. Capture (Source) Site Configuration
5. Create Streams Environment
Data Replication Using Streams

- **10.2**
  - Use Streams on Maintenance page of OEM

- **10.1.0.4**
  - Use “classic OEM” to manage Oracle Streams (10.1.0.2.0 Client)
    - Alters database for Supplemental Logging
    - Needs Oracle Management Server for exp/imp task
    - Saved scripts can be reviewed and run manually

- Write your own scripts
Database Configuration

Initialization Parameters:

- **AQ_TM_PROCESSES** => 1  # For User Events Only
- **COMPATIBLE** = 10.1.0  # Or higher
- **GLOBAL_NAMES** = TRUE
- **JOB_QUEUE_PROCESSES** = 2  # Or more
- **LOGMNR_MAX_PERSISTENT_SESSIONS** => # of Capture Processes
- **OPEN_LINKS** = 4  # Or more
- **PARALLEL_MAX_SERVERS** = <?>
- **PROCESSES** = <?>
- **SHARED_POOL_SIZE** = <?> 100MB Min + 10MB/Capture
- **STREAMS_POOL_SIZE** = > 200 MB
- **TIMED_STATISTICS** = TRUE
Database Configuration

- Initialization Parameters:
  - For Downstream Capture
    - LOG_ARCHIVE_DEST_n = (1..10)
    - LOG_ARCHIVE_DEST_STATE = ENABLE
    - REMOTE_ARCHIVE_ENABLE = TRUE
Streams Administrator Account

- Create default tablespace for Streams Administrator Account
  - Stores the queue table

```sql
CREATE TABLESPACE streams_queue_ts DATAFILE
  '/u19/oradata/DBXT/streams_queue_ts_01.dbf' SIZE 1000M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512k
  SEGMENT SPACE MANAGEMENT AUTO;
```
Streams Administrator Account

- **Create Streams Administrator Account**
  - To Configure/Manage/Administer Streams Environment:
    - Capture Site
    - Propagation Site
    - Apply Site

CREATE USER STRMADMIN IDENTIFIED BY secret101
DEFAULT TABLESPACE streams_queue_ts
QUOTA UNLIMITED ON streams_queue_ts;

GRANT CONNECT, RESOURCE, DBA to STRMADMIN;
GRANT SELECT ANY DICTIONARY TO STRMADMIN;
EXEC DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE('STRMADMIN');
Streams Administrator Account

- Create Streams Queues
  - At Source (Capture) and Target (Apply) Site

CONNECT STRMADMIN/&strmadmin_pwd

BEGIN
  DBMS_STREAMS_ADM.SET_UP_QUEUE(
    queue_table => 'STREAMS_QUEUE_TABLE',
    storage_clause => 'TABLESPACE STREAMS_QUEUE_TS',
    queue_name => 'DBXT_TO_DBXP',
    queue_user => 'STRMADMIN');
END;
/

Target Site Configuration

Create APPLY process, rule and rule set

CONNECT STRMADMIN /&strmadmin_pw
BEGIN
   DBMS_STREAMS_ADMIN.ADD_TABLE_RULES(
      table_name    => 'STRMDEMO.TEST',
      streams_type  => 'APPLY',
      streams_name  => 'STREAMS_APPLY',
      queue_name    => 'STRMADMIN.DBXT_TO_DBXP',
      include_dml   => TRUE,
      include_ddl   => TRUE,
      source_database => 'DBXT.world');
END;
/

Define a user name for the Apply process to use to apply events to database objects.

CONNECT / AS SYSDBA
BEGIN
    DBMS_APPLY_ADM.ALTER_APPLY(
        apply_name => 'STREAMS_APPLY',
        apply_user => 'STRMADMIN');
END;
/

The Apply user must have all grants/permissions to perform DDL and DML operations on the objects.
Target Site Configuration

- Start the APPLY process

    CONNECT STRMADMIN/&strmadmin_pw

    BEGIN
        DBMS_APPLY_ADM.START_APPLY(
            apply_name => 'STREAMS_APPLY');
    END;

    /
Source Site Configuration

Database Link

REM – Create the Streams Administrator Account at the Apply site, and then create a private database link at the Capture site

CONNECT STRMADMIN/&strmadmin_pw

create database link DBXP
connect to strmadmin identified by &strmadmin_pw_applysite using 'DBXP.world' /


Source Site Configuration

- **Enable Supplemental Logging**
  - All tables in Streams setup at Capture Site
  - Use columns in PK constraints for Supplemental log group
    
    ```sql
    ALTER TABLE TEST
    ADD SUPPLEMENTAL LOG GROUP SLGPK_TEST
    (ACCOUNT_NUM) ALWAYS;
    ```

  - If no PK, then unique key columns can be used.
Source Site Configuration

- Disable NOLOGGING operations

  - You cannot allow ‘nologging’ operations on objects participating in Streams

- Enable FORCE LOGGING at tablespace or database level

  ```sql
  ALTER DATABASE FORCE LOGGING;
  ALTER TABLESPACE <tablespace_name> FORCE LOGGING;
  ```
Source Site Configuration

● Create Capture process, Rule and Rule set

CONNECT STRMADMIN/&strmadmin_pw

BEGIN

DBMS_STREAMS_ADM.ADD_TABLE_RULES(
    table_name => 'STRMDEMO.TEST',
    streams_type => 'CAPTURE',
    streams_name => 'STREAMS_CAPTURE',
    queue_name => 'STRMADMIN.DBXT_TO_DBXP',
    include_dml => TRUE,
    include_ddl => TRUE,
    source_database => 'DBXT.world');

END;
/

Source Site Configuration

Create Propagation process, Rule and Rule set

CONNECT STRMADMIN/&strmadmin_pw

BEGIN

    DBMS_STREAMS_ADM.ADD_TABLE_PROPAGATION_RULES(
        table_name => 'STRMDEMO.TEST',
        streams_name => 'STREAMS_PROPAGATE',
        source_queue_name => 'STRMADMIN.DBXT_TO_DBXP',
        destination_queue_name => 'STRMADMIN.DBXT_TO_DBXP@DBXP.world',
        include_dml => TRUE,
        include_ddl => TRUE,
        source_database => 'DBXT.world');

END;
/

31
Create Streams Environment

- **Object Instantiation**
  - Creating the objects at the Apply site based on the objects at the Capture site.
  - Populating the Streams data dictionary with metadata.
  - Setting the instantiation SCN for the streamed objects at Apply site.

- For tables, instantiation can be done using export/import.
Create Streams Environment

Object Instantiation

- Export table with \texttt{OBJECT\_CONSISTENT=Y}
- Import table with \texttt{STREAMS\_INSTANTIATION=Y}
- Use \texttt{ROWS=Y (or N)} and \texttt{IGNORE=Y (or N)}
- Manual Instantiation using SCN
  - Select \texttt{CURRENT\_SCN} from \texttt{V\$DATABASE} at Capture Site
  - Use this SCN to instantiate tables at APPLY site:

\begin{verbatim}
EXEC DBMS\_APPLY\_ADM.SET\_TABLE\_INSTANTIATION\_SCN( -
  source\_object\_name => 'STRMDEMO.TEST', -
  source\_database\_name => 'DBXT.WORLD', -
  instantiation\_scn => & 'SCN');
\end{verbatim}
Create Streams Environment

- At Source (Capture) Site
  - Start the Capture Process

```sql
CONNECT STRMADMIN/&strmadmin_pw

BEGIN
  DBMS_CAPTURE_ADM.START_CAPTURE(
    capture_name => 'STREAMS_CAPTURE');
END;
/
```
Create Streams Environment

- Streams Environment is now ready.

- DDL and DML changes made to TEST table at the Capture site (DBXT.world) will be replicated to TEST table at Apply site (DBXP.world)
Streams - Turbulences

- CAPTURE process may need to read an older archived log due to how logminer captured checkpoints.

- Change CAPTURE parameters
  - DBMS_CAPTURE_ADM.SET_PARAMETER procedure:
    - _CHECKPOINT_FREQUENCY = 1
    - _CHECKPOINT_FORCE = Y
    - _SGA_SIZE > 10M (default)
Monitoring Streams

- DBA_APPLY_ERROR
- DBA_APPLY
- DBM_CAPTURE
- V$STREAMS_APPLY_COORDINATOR
- V$STREAMS_APPLY_READER
- V$STREAMS_APPLY_SERVER
- OEM Screens
- streams_health_check.sql
- strmmon Utility
Periodically force logminer checkpoint

Ensures CAPTURED_SCN and APPLIED_SCN in DBA_CAPTURE are maintained correctly

```
CONN STRMADMIN/&pw
BEGIN
    DBMS_CAPTURE_ADM.SET_PARAMETER(
    CAPTURE_NAME => 'STREMS_CAPTURE',
    PARAMETER    => '_CHECKPOINT_FORCE',
    VALUE        => 'Y');
END;
/
```
Maintenance

- **Reset FIRST_SCN value in DBA_CAPTURE**
  - Deletes old rows from various LOGMNR* tables. Use DBMS_CAPTURE_ADM.ALTER_CAPTURE procedure to change FIRST_SCN.

```
CONN STRMADMIN/&pw
BEGIN
    DBMS_CAPTURE_ADM.ALTER_CAPTURE(
        CAPTURE_NAME => 'STREAMS_CAPTURE',
        FIRST_SCN => &new_first_scn);
END;
/
```
Streams Resources

- Oracle Metalink
  - Knowledge
    - Database (Support Categories)
      - Information Integration
      - Streams

- Oracle Streams Manuals
  http://www.tahiti.com
  http://otn.oracle.com/pls/db102/db102.federated_search
Streams Resources

- Oracle Technology Network
  - Streams Forum

http://forums.oracle.com/forums/forum.jsp?forum=70

Oracle 10g Streams: Data Replication Made Easy

Questions ?????

Kirtikumar_Deshpande@yahoo.com