Agenda

• Available High Availability Solution
• What is Oracle 9i Streams
• Architecture of Oracle Streams
• Common terminology
• Oracle HA Features Comparison
• Pre-requisite for Streams Implementation
• Common Streams Element DD Views
• Questions & Answers
Available Oracle High Availability Solutions

- Real Application Cluster
- Oracle Advance Replication
- Oracle Data Guard [Physical/Logical Standby]
- Oracle Failsafe
- Oracle 9i Streams
What is Oracle STREAMS

• New Oracle 9i feature to share information between Oracle as well as non-Oracle databases.

• Streams is a set of processes and database structures that allow sharing of data and messages in a data stream.

• The information placed in the data stream is called Event which can be
  → DML/DDL changes Or
  → User created messages

• Streams Uses Advanced Queues and Log Miner.
Streams Contains 3 basic elements that enables you to control

- What Information is put into the stream
  - [Capture]
- How the Stream Flow from node to Node
  - [Stage & Propagate]
- What Happens to events as they flow into each node
  - [Optional Transformation]
- How the Stream terminates
  - [Apply]
RULES

• A Rule is a database object that describes what information need to be share and where to share it.

• Rules are evaluated by Oracle9i's built-in rules engine and evaluate to a Boolean result (TRUE or FALSE)

• Rules can be used during capture, propagate and apply process

• Rules can be defined at different levels such as table level, schema level and database level.

• You can group related rules together into rule sets. A rule can be part of multiple rule set or no rule set.

• Use DBMS_RULE_ADM procedure
CREATE_RULE_SET / CREATE_RULE / ADD_RULE
Logical Change Record

• An LCR is an object with a specific format that describes a database change. LCRs are of two types: row LCRs and DDL LCRs.

• A row LCR describes a change to the data in a single row or a change to a single LOB column in a row as a result of DML.

• A DDL LCR describes a data definition language (DDL) change.

• Each LCR [DDL or DML] contains the following main information:
  - The name of the source database where the DDL/DML change occurred.
  - The type of DDL/DML statement like Insert/Update/Alter table.
  - The schema name of the user.
  - The name of the database object.
  - The SCN when the change was written to the redo log.
Elements of Streams

- Capture Process
- Staging Process
- Propagation
- Transformation
- Apply Process
CAPTURE PROCESS

- Reads the Redo logs
- Extracts the DDL/DML as per predefined set of RULES which define what changes to be captured
- Format the information into events also called LCR
- Place the information in the queue also called Staging
Capture Process - Log Miner

- A capture process uses LogMiner Infrastructure to capture database changes. Streams configures LogMiner automatically.
- By Default Logminer Tables are created to use SYSTEM tablespace and is not recommended.
- Re-create logminer tables in different tablespace before configuring stream setup

SQL> Execute
DBMS_LOGMNR_D.SET_TABLESPACE('<TblSpNam>');

- If using OEM to configure Stream environment, then it will check and prompt your for different tablespace
- Oracle 10g contain SYSAUX tablespace to store all auxiliary metadata related to Oracle options like Streams.
Capture Process – Redo Log

- Capture process Reads either online Redo log file or Archived redo log files
- Archived Files are Used in Oracle 9i RAC or at time when there is high DML activity on the Database
- Seamless transition from reading an online redo log to reading an archived redo log and vice versa
- Oracle 10g support Online Redo log capture for Oracle RAC
- Oracle 10g support mining the archived logs of the source database at an alternative database, assuming the alternative database is on a similar platform type and operating system
Capture Process - Creation

BEGIN

DBMS_STREAMS_ADM.ADD_TABLE_RULES(
    table_name => 'hr.employees',
    streams_type => 'capture',
    streams_name => 'strm01_capture',
    queue_name => 'strm01_queue',
    include_dml => true,
    include_ddl => true,
    include_tagged_lcr => false);

END
Capture Process - Components

Read redo records. Merge redo from different threads (Read) → Transform redo records into Logical Change Records (LCR) (Prepare) → Group LCRs into txns. Fix chained rows /LOBs /Handle DDLs etc (Build)

Redo Stream

The CPnn process performs the final rule-evaluation on the LCRs and then enqueued LCR into staging area

- It depends on **PARALLELISM** setting.
- If parallelism is set to a value of 3 or greater
- If **parallelism = 5**, then a capture process uses one reader server, three **Prepare** servers, and one builder server.
Capture Process - Parallelism

BEGIN

DBMS_CAPTURE_ADM.SET_PARAMETER(
  capture_name => 'stream 01_capture',
  parameter => 'parallelism',
  value => '3');

END;

- If parallelism is set to 2 or lower, then a capture process itself [cp nn (01-99)] performs all the work without using any parallel execution servers.
Capture Process

– A capture process never captures changes in the SYS and SYSTEM schemas
– A capture process does not capture DBMS_REDEFINITION package changes
– A capture process uses queue buffers available in shared pool area unlike queue tables on disk in AQ
– You can create, alter, start, stop, and drop a capture process

BEGIN
  DBMS_capture_ADM.Start_capture
  (Capture_name=>’capture_hr’);
END
<table>
<thead>
<tr>
<th>Supported Data types</th>
<th>Unsupported Data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR, NCHAR</td>
<td>NCLOB</td>
</tr>
<tr>
<td>VARCHAR2, NVARCHAR2</td>
<td>LONG</td>
</tr>
<tr>
<td>NUMBER</td>
<td>LONG RAW</td>
</tr>
<tr>
<td>DATE</td>
<td>BFILE</td>
</tr>
<tr>
<td>CLOB, BLOB</td>
<td>ROWID</td>
</tr>
<tr>
<td>RAW</td>
<td>UROWID</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>User-defined types</td>
</tr>
<tr>
<td>TIMESTAMP WITH TIME ZONE</td>
<td>• Object types</td>
</tr>
<tr>
<td>TIMESTAMP WITH LOCAL TIME ZONE</td>
<td>• REFS</td>
</tr>
<tr>
<td>INTERVAL YEAR TO MONTH</td>
<td>• Varrays</td>
</tr>
<tr>
<td>INTERVAL DAY TO SECOND</td>
<td>• Nested tables</td>
</tr>
</tbody>
</table>
# STREAMS Non-Supported DDL Operations

<table>
<thead>
<tr>
<th>Create or Alter Database</th>
<th>Alter Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create/Alter/Drop Rollback</td>
<td>Alter System</td>
</tr>
<tr>
<td>Create/Alter/Drop Type</td>
<td>Truncate</td>
</tr>
<tr>
<td>Create/Alter/Drop Profile</td>
<td>Set Role</td>
</tr>
<tr>
<td>Create/Drop Library</td>
<td>Set Transaction</td>
</tr>
<tr>
<td>Create/Drop Directory</td>
<td>Set Constraint</td>
</tr>
<tr>
<td>Create Control File</td>
<td>Analyze</td>
</tr>
<tr>
<td>Create Spfile</td>
<td>Explain</td>
</tr>
<tr>
<td>Create Pfile</td>
<td>Call</td>
</tr>
<tr>
<td>Lock Table</td>
<td>PL/SQL Procedural Calls</td>
</tr>
</tbody>
</table>
# DDL Captured but Not applied

<table>
<thead>
<tr>
<th>CREATE/ALTER/DROP MATERIALIZED VIEW LOG</th>
<th>CREATE SCHEMA AUTHORIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE , ALTER, or DROP MATERIALIZED VIEW</td>
<td>CREATE or DROP DATABASE LINK</td>
</tr>
<tr>
<td>CREATE or ALTER TABLE for Index-organized tables</td>
<td>RENAME (use ALTER TABLE instead)</td>
</tr>
<tr>
<td>CREATE TABLE AS SELECT for clustered tables</td>
<td></td>
</tr>
</tbody>
</table>
Elements of Streams

- Capture Process
- **Staging Process**
- Propagation
- Transformation
- Apply Process
STAGING PROCESS

• It is a queue that provides a service to store and manage captured events.
• Message remain in staging area until consumed by all subscribers.
• If the subscriber is another staging area, the event is propagated to the other staging area, either within the same database or in a remote database.
There are two types of events that can be staged in a Streams queue:

- logical change records (LCRs) and
- User messages.

Your applications can enqueue/dequeue user messages using PL/SQL (DBMS_AQ package), JMS, OCI.

Staged events can be consumed or propagated, or both.
Elements of Streams

- Capture Process
- Staging Process
- Propagation
- Transformation
- Apply Process
PROPAGATION

- Streams uses job queues to propagate events using job queue processes (J nnn)
- You can CREATE/DROP a propagation Using DBMS_STREAMS_ADM DBMS_PROPAGATION_ADM package
- The default schedule has the following properties:
  - The start time is SYSDATE().
  - The duration is NULL, which means infinite.
  - The next time is NULL
- You can alter the schedule for a propagation with ALTER_PROPAGATION_SCHEDULE procedure in the DBMS_AQADM package.
BEGIN
    Dbms_Streams_Adm.Add_Table_Propagation_Rules(
        table_name => 'hr.departments',
        streams_name => 'strm01_propagation',
        source_queue_name => 'strmadmin.strm01_queue',
        destination_queue_name =>
            'strmadmin.strm02_queue@dbs2.net',
        include_dml => true,
        include_ddl => true,
        include_tagged_lcr => false,
        source_database => 'dbs1.net'
    );
END;
Directed Network

- Queue Forwarding and Apply Forwarding
Elements of Streams

- Capture Process
- Staging Process
- Propagation
- **Transformation**
- Apply Process
Transformations can be performed:
- as events enter the staging area
- as events leave the staging area
- as events propagate between staging areas

Transformation examples:
- change format, data type, column name, table name
Elements of Streams

- Capture Process
- Staging Process
- Propagation
- Transformation
- Apply Process
APPLY PROCESS

• Oracle background process [ ap nn] that dequeue LCRs and user messages
• For non-LCR messages, the apply servers pass the events to the message handler.
• Automatic conflict detection with optional resolution → unresolved conflicts placed in exception queue
- Separate Apply process for each Capture Process
- Separate Apply Process for LCR and User Events
Apply Process - Components

- Dequeue Events and returns the assembled transaction to the Coordinator (Read)
- Gets transactions from Reader and pass it among Apply servers based on dependency (Coordinator)
- One or More Apply server that apply LCR or pass the LCRs to appropriate handlers (Apply Server)

Staging Queue

PARALLELISM = # of above Apply Server

PRNewswire
United Business Media
BEGIN

DBMS_STREAMS_ADM.ADD_TABLE_RULES(
  table_name => 'hr.employees',
  streams_type => 'apply',
  streams_name => 'apply_emp',
  queue_name => 'strmadmin.streams_queue',
  include_dml => true,
  include_ddl => false,
  source_database => 'cpap.net');

END;
APPLY PROCESS - Start

BEGIN

DBMS_APPLY_ADM.SET_PARAMETER(
  apply_name => 'apply_emp',
  parameter => 'disable_on_error',
  value => 'n');

END;
/

BEGIN

DBMS_APPLY_ADM.START_APPLY(
  apply_name => 'apply_emp');

END;
**APPLY PROCESS - Commit**

Apply servers may apply transactions at the destination in an order that is different from the commit order at the source:

```sql
BEGIN
  DBMS_APPLY_ADM.SET_PARAMETER(
    apply_name => 'strm01_apply',
    parameter => 'commit_serialization',
    value => 'none');
END;
```

**Commit Serialization** has the following value:

- **full**: Default and order is same as at source database
- **none**: Commit transactions in any order. Performance is best if you specify this value.
More Facts

Missing Columns at the Destination Database

Apply process raises an error and moves the transaction into an exception queue.

**Fix** :- Creating a rule-based transformation or DML handler that eliminates the missing columns from the LCRs before they are applied.

Column Data type Mismatch

Apply process places transactions into an exception queue.

**Fix** :- Create a rule-based transformation or DML handler that converts the data type.
Extra Columns at the Destination Database

• Apply process check for Dependency for the column and If the extra columns are not used for dependency computations, then applies changes to the destination table.

• If column defaults exist for the extra columns, then these defaults are used for these columns for all inserts.
Avoid System Generated Names

– For example, DDL statement at a source database:
  
  `CREATE TABLE EMP (n1 NUMBER NOT NULL);`

– This results in a NOT NULL constraint with a system-generated name. For example, sys_001500.

– When DDL is applied at a destination, the system-generated name for this constraint may be sys_c1000.

– Again DDL statement at the source database:
  
  `ALTER TABLE EMP DROP CONSTRAINT sys_001500;`

– It fails at the destination database during the apply process and so Fix is

  `CREATE TABLE EMP (n1 NUMBER CONSTRAINT emp_null_nn NOT NULL);`
Oracle → Non Oracle Replication

Diagram showing the process of Oracle to Non Oracle Replication:
- Oracle Database
  - Queue
    - Dequeue Events
    - Apply Process
  - Heterogeneous Services
- Non-Oracle Database
  - Apply Changes
  - Gateway
  - Database Objects
Oracle → Non Oracle Replication

- Parallel apply to non-Oracle databases is not supported.
- Error handlers and conflict handlers are not supported.
- If an apply error occurs, then the transaction moved into an exception queue in the Oracle database.
- The apply process detects data conflicts but automatic conflict resolution is not supported. Therefore, any data conflicts encountered are treated as apply errors.
- The apply process cannot apply DDL changes at non-Oracle databases.
Non-Oracle → Oracle Replication
## Feature Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>Advanced Replication</th>
<th>RAC</th>
<th>Data Guard</th>
<th>Streams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire Database Replication</td>
<td>YES</td>
<td>N/A</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Schema Replication</td>
<td>YES</td>
<td>N/A</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Table Replication</td>
<td>YES</td>
<td>N/A</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>DML / DDL Replication</td>
<td>YES</td>
<td>N/A</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Instance Redundant</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Database Redundant</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Cluster Software</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Data type Support</td>
<td>SOME</td>
<td>ALL</td>
<td>SOME</td>
<td>SOME</td>
</tr>
<tr>
<td>Heterogeneous Db Support</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>
## Feature Comparison

<table>
<thead>
<tr>
<th></th>
<th>Advanced Replication</th>
<th>RAC</th>
<th>Data Guard</th>
<th>Streams</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OS between Source &amp; Target</strong></td>
<td>Can be Different</td>
<td>Must be Same</td>
<td>Must be Same</td>
<td>Can be Different</td>
</tr>
<tr>
<td><strong>Oracle S/w Version between Source and Target</strong></td>
<td>Can be Different</td>
<td>Must be Same</td>
<td>Must be Same</td>
<td>Can be Different [at least 9.2]</td>
</tr>
<tr>
<td><strong>Hardware Location</strong></td>
<td>N/A</td>
<td>Same Place</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Cost Factor for Licensing</strong></td>
<td>Included</td>
<td>Extra Cost</td>
<td>Included</td>
<td>Included</td>
</tr>
</tbody>
</table>
Pre-requisite for Streams

- Oracle Software Version 9.2.0.3 or higher
- Database should be in ARCHIVELOG mode
- Override No logging operations by using
  ```sql
  ALTER DATABASE/TABLESPACE Force Logging;
  ```
- Following init.ora parameter setting
  - AQ_TM_PROCESSES to be at least 1
  - COMPATIBLE to be 9.2.0 or higher
  - GLOBAL_NAMES=true for sharing information between databases
  - JOB_QUEUE_PROCESSES to be at least 2
  - SHARED_POOL_SIZE increase by 10MB
Important DD Views - CAPTURE

DBA_CAPTURE
DBA_CAPTURE_PARAMETERS
DBA_CAPTURE_PREPARED_DATABASE
DBA_CAPTURE_PREPARED_SCHEMA
DBA_CAPTURE_PREPARED_TABLES
V$STREAM_S_CAPTURE
Important DD Views - STAGE

DBA_QUEUES
DBA_QUEUE_PUBLISHERS
DBA_QUEUE_TABLES
AQ $<queue TableName> - Enqueue & Dequeue Information
Important DD Views - PROPAGATE

DBA_DB_LINKS
DBA_JOBS
DBA_JOBS_RUNNING
DBA_PROPAGATION
DBA_QUEUE_SCHEDULES
Important DD Views - APPLY

DBA_APPLY
DBA_APPLY_PROGRESS
DBA_APPLY_PARAMETERS
DBA_APPLY_CONFLICT_COLUMNS
DBA_APPLY_DML_HANDLERS
DBA_APPLY_ERROR
DBA_APPLY_INSTANTIATED_OBJECTS
DBA_APPLY_KEY_COLUMNS
V$STRAMS_APPLY_COORDINATOR
V$STRAMS_APPLY_READER
V$STRAMS_APPLY_SERVER
Questions & Answers