Oracle Data Guard Logical Standby Database

Paul Baumgartel
Transcentive, Inc.
New York Oracle Users Group
September 23, 2003

Introduction

- Data Guard is major component of Oracle's High Availability (HA) facility
- Other features include Real Application Clusters (RAC), Advanced Replication, and Oracle Streams
- Physical standby database available since Oracle8
- Logical standby introduced in 9iR2

Choosing a standby database configuration

- Physical standby best for disaster recovery
  - Most efficient
  - Less complex
- Logical standby more flexible
  - Not everything must be duplicated
  - Can be used at all times for reporting

Physical standby concepts

- Exact file copy of primary database
- Activity on primary duplicated via application of redo information
- Log transport services transmit in LOG or ARCH mode
- Standby is mounted but not open

Physical standby concepts

- Log apply services receive and apply redo directly from logs
- Most efficient due to use of low-level recovery mechanism (database block change vectors)
- No user activity while in recovery mode
- May be opened for read-only user activity after suspending recovery

Logical standby concepts

- Standby starts as, but need not remain, exact copy
- May contain schemas and objects that do not exist on primary
- Log transport services transmit in LOG or ARCH mode
- Standby is open for read/write activity (but no writes on standby objects)

Logical standby concepts

- Log apply services use LogMiner to construct SQL from redo log contents, then apply it to standby objects
- DDL may be included in or excluded from log apply stream
- Progress of log apply services may be monitored

Logical standby: unsupported data types

- NCLOB
- LONG
- LONG RAW
- BFILE
- ROWID
- UROWID

Logical standby: unsupported objects

- user-defined types
- object types REFs
- varrays
- nested tables
- **Unsupported Tables, Sequences, and Views**
- User-defined tables and sequences in the SYS schema
- Tables with unsupported data types
- Tables using data segment compression
- Index-organized tables

Preparing the primary database

- Enable forced logging
- Enable archiving and define local archive destination
- Set `LOG_PARALLELISM` to 1
- Ensure rows can be uniquely identified (query `DBA_LOGSTDBY_NOT_UNIQUE`)

Preparing the primary database

- Enable supplemental logging
- Perform log switch
- Create alternate tablespace for LogMiner objects

Creating the standby database

- Copy the primary
- Move files to the standby machine
- Rename data files
- Turn on Database Guard

Creating the standby database

- Rename the database
- Register first archived log to be applied
- Start log apply services
- Enable archiving to standby

Verify standby database

- DBA_LOGSTDBY_LOG
- V$LOGSTDBY
- DBA_LOGSTDBY_PROGRESS
- These views show available logs, progress of log application, and snapshots of current SQL activity
- Transaction information and SCNs

Management of logical standby

- Exclude objects or schemas from log application
- Describe operations that should not be applied to standby
- Log errors and continue apply process
- Database guard: `ALL`, `STANDBY`, `NONE`

Management of logical standby

- Temporarily bypass guard to modify objects
- Constraints and triggers enabled but not executed
- Delay log application to avoid error duplication

Logical standby performance

- Indexes on tables with no primary key
- `TRANSACTION_CONSISTENCY: FULL, READ_ONLY, NONE`
- Adjust `PARALLEL_MAX_SERVERS`
- Adjust `SHARED_POOL_SIZE` to improve performance of LogMiner components

Limitations

- Logical standby not compatible with Oracle Label Security
- Bugs in log apply process (fixed on Unix platforms, not yet fixed on Windows)
- Performance can be poor due to nature of process
- Deletion of 1000 rows is represented in logs by 1000 change vectors; LogMiner extracts and creates 1000 `DELETE` statements

Summary

- Logical standby worthwhile Data Guard option
- Requires careful setup and monitoring
- One of growing number of Oracle HA alternatives including replication, Data Guard, RAC, and Streams

Contact information

- paul.baumgartel@aya.yale.edu
- 917 549-4717

- Feel free to get in touch!