

Downstream Database Configuration

The objective of this configuration is to isolate the source database against network or database problems at the destination sites.

Downstream Capture

The capture process runs on a different database the source database, the downstream database. Archived redo log files from the source database are copied to the downstream database where the capture process captures the changes. The files can be copied using log transport services or even manually.

Advantages

- Use of fewer resources on the source database because the downstream database performs most of the required work
- Copying redo log files to one or more downstream databases provides improved protection against data loss

Disadvantages

- Definite latency in the replication process
- Downstream database is an additional expense

Restrictions / Requirements

- Source database must be running at least Oracle Database 10g and the downstream capture database must be running the same version as the source database or higher
- Operating system on the source and downstream capture sites must be the same (release and directory structure can be different)
- Hardware architecture on the source and downstream capture sites must be the same

Creating a Real Time Downstream Capture Environment

- **How to configure Streams Real-Time Downstream Environment - metalink note 753158.1**

Preparing to Copy Redo Data

The following initialization parameters must be set to configure the redo transport services. (Downstream database is dwsdb.cern.ch)

- At the source database:

```
LOG_ARCHIVE_DEST_2 = 'SERVICE=dwsdb LGWR ASYNC OPTIONAL
                     NOREGISTER VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_
LOG_ARCHIVE_DEST_STATE_2 = 'ENABLE'
LOG_ARCHIVE_CONFIG='DG_CONFIG=(sourcedb, dwsdb)'
```

- At the downstream database:

```
LOG_ARCHIVE_DEST_2 = 'LOCATION=/ORA/dbs02/oracle/admin/arch/logfrom_sourcedb/
                     MANDATORY VALID_FOR=(STANDBY_LOGFILE,PRIMARY_ROLE) '
LOG_ARCHIVE_DEST_STATE_2 = 'ENABLE'
LOG_ARCHIVE_CONFIG='DG_CONFIG=(sourcedb, dwsdb)'
```

DownstreamDatabaseConfiguration < PSSGroup < TWiki

- Create the standby redo log files at the downstream database:
 - ◆ Determine the log file size used on the source database and the number of log file groups

```
SQL> select THREAD#, GROUP#, BYTES/1024/1024 from V$LOG;
```

- ◆ If source is NON-RAC:

THREAD#	GROUP#	BYTES
1	1	104857600
1	2	104857600
1	3	104857600

◇ Previous output indicates that we got 3 groups each with size 100M. This means that we will need to have 4 standby-log groups with at least 100M each.

◇ Add the standby log file groups to the downstream database

```
ALTER DATABASE ADD STANDBY LOGFILE GROUP 4
('/path/sdwsdb_redo_04a.log', '/path/sdwsdb_redo_04b.log')
SIZE 100M;
....
ALTER DATABASE ADD STANDBY LOGFILE GROUP 7
('/path/sdwsdb_redo_07a.log', '/path/sdwsdb_redo_07b.log')
SIZE 100M;
```

- ◆ If source is RAC:

THREAD#	GROUP#	BYTES
1	1	104857600
1	2	104857600
2	3	104857600
2	4	104857600

◇ Previous output indicates that we got two THREADS(instances) each one has two redo-log groups each one have a size 100M. This means that we will need to have 3 standby-log groups per THREAD with each group at least 100M.

◇ Add the standby log file groups to the downstream database

```
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 11 SIZE 100M;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 12 SIZE 100M;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 GROUP 13 SIZE 100M;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 21 SIZE 100M;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 22 SIZE 100M;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 2 GROUP 23 SIZE 100M;
```

Creating Downstream Capture process

The attached document `setupDownstreamRealTime_example.sql` contains an example of setting up a Streams environment using Real Time Downstream capture.

You can find more information on the **Oracle® Streams Concepts and Administration 10g**.

Split & Merge solution

The objective is to isolate the replicas sites against each other. When one site goes down, the source queue is

DownstreamDatabaseConfiguration < PSSGroup < TWiki

filled with the LCRs/events which have not been consumed by the down site, affecting to the performance of the whole configuration. In order to avoid that, the following solution is proposed:

- The idea is to split the capture process in two capture processes:
 - ◆ The original one (real-time downstream capture in our setup) for the sites working well
 - ◆ New capture process for the site/s unstable/s
 - ◇ New capture queue and new propagation job are defined
 - ◇ The original propagation job is dropped so the spilled LCRs are removed from the original source queue
- Once the site is stable again, both capture processes are merged in the original one

(see the attached script for more details on how to implement this solution)

- Solution tested successfully with the help of **Volker Kuhr** (Oracle consulting)

This topic: PSSGroup > DownstreamDatabaseConfiguration

Topic revision: r5 - 2009-03-06 - EvaDafonte



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback