

Hello DBA,

Myself **Rahul Kurkute**. Typically, for each production database, we establish a standby database and conduct disaster recovery drills, to test the disaster recovery sites. Many companies periodically run their workloads on the disaster recovery site and then switch back to the primary site. It's essential for companies to have a robust disaster recovery plan to ensure they can resume operations quickly. I've observed that many DR drills take over **90 minutes** to complete.

This brings us to **RTO and RPO**. In short, **RTO (Recovery Time Objective)** refers to the amount of time needed to restore your critical infrastructure (In our case database) after a disaster. On the other hand, **RPO (Recovery Point Objective)** represents the maximum acceptable period between a disaster and your most recent backup, indicating the potential data loss. For more information, try searching on Google—there are plenty of great blogs available.

RTO can be used not only for disaster scenarios but also during pre-planned SwitchBack (SB) and SwitchOver (SO) activities.

In this document, I will outline a straightforward simple method to shorten DR drill time and enhance **RTO**.

For RTO optimization, it's important to configure the lesser known but highly useful **STANDBY_DB_PRESERVE_STATES** parameter on the standby database.

The **STANDBY_DB_PRESERVE_STATES** parameter in Oracle Database is used in conjunction with Oracle Data Guard to enhance the efficiency of disaster recovery operations. It plays a role in preserving the state of a standby database during certain operations, which can help reduce the RTO (Recovery Time Objective).

From **Oracle Database 12c Release 2 (12.2.0.1)**, when a physical standby database is converted into a primary you have the option to keep any sessions connected to the physical standby connected, without disruption during the switchover/failover cases. To enable this feature, set the STANDBY_DB_PRESERVE_STATES initialization parameter in your init.ora file before the standby instance is started or It requires standby database bounce.

By this parameter, Active user sessions on the standby database are preserved during a switchover Activity thereby reducing the time needed for application sessions to reconnect.

What STANDBY_DB_PRESERVE_STATES Does???

The STANDBY_DB_PRESERVE_STATES parameter controls the preservation of the sessions state of the standby database during a switchover or failover operation. When this parameter is set, Oracle Data Guard maintains the standby database's state information, which can facilitate faster recovery and switchover operations.

The STANDBY_DB_PRESERVE_STATES controls whether user sessions and other internal states of the instance (Buffers) are retained when a readable physical standby database is converted to a primary database. This parameter is meaningful on a physical standby database that is open in real-time query mode (READ ONLY MODE).

You can use below command to set the STANDBY_DB_PRESERVE_STATES parameter.

The same value must be used on all instances.

```
=====
ALTER SYSTEM SET STANDBY_DB_PRESERVE_STATES=SESSION SCOPE=SPFILE SID='*';
=====
```

The possible values for the parameter are NONE, SESSION, and ALL.

NONE:

Nothing is retained. All sessions are disconnected and all buffers are flushed. This is the default value.

SESSION:

User sessions are retained. When the database is reopened as the primary, the retained sessions resume their operations as if nothing had happened. If the database (or an individual PDB) is not opened in the primary role, the sessions will be terminated.

BUFFER:

All current buffers are retained. Media recovery buffers are converted to current buffers and retained, if possible. When the database is reopened as the primary, the retained buffers are available for use, which may enable queries to run faster during the first few minutes of operation. If the database (or an individual PDB) is not opened in the primary role, the buffers will be flushed.

ALL:

This value is equivalent to setting both the SESSION and BUFFER values.

Reducing RTO isn't just about configuration. it's essential to have a well-defined action plan. Without a proper plan, achieving a lower RTO won't be possible.

Action Plan:

Before DR drill activity we need to follow below steps to save DR time.

1. Set parameter STANDBY_DB_PRESERVE_STATES in database.
`ALTER SYSTEM SET STANDBY_DB_PRESERVE_STATES=SESSION SCOPE=SPFILE SID='*';`
2. Restart Standby Database in **READ ONLY WITH APPLY MODE**
3. Start Application / JVM/ OHS on Standby Side. Make sure the Application is pointing to DR database only, Test connection for data sources and ensure all deployments are in active mode.

During Switchover Activity make sure Primary Database in READ WRITE mode and SO status showing "TO_STANDBY"

1. **Make sure Standby Database in READ ONLY mode.**

2. **Primary Side:**

```
select NAME,OPEN_MODE,SWITCHOVER_STATUS,DATABASE_ROLE from gv$database;
alter database switchover to <db_unique_name> verify;  <-- To verify switchover activity.
alter database switchover to <db_unique_name>;
```

3. **Standby Site:**

```
alter database open; <--- This need to be run on all instances
```

4. Now check if Application / JVM / OHS is up and running fine or not without restart.

Key Notes:

- 1) Sessions that have long running queries or are using database links will not be retained regardless of the setting of this parameter.
 - 2) If Primary DB having running Stats gather job, then it won't continue on standby database as standby database run with read only mode.
 - 3) Check with respected team and take necessary approval before implementation.
 - 4) When the standby_db_preserve_states parameter is set to BUFFER (or ALL), there is a risk of encountering a bug in your environment. To prevent the problems from occurring in the future, set standby_db_preserve_states to SESSION or NONE and monitor the environment very closely.
 - 5) As of Oracle Database 18c, the database buffer cache state is maintained on an Active Data Guard standby during a role transition so that application performance is not affected by physical blocks read from disk to populate the buffer cache. This results in improved application performance on the new primary after a role transition.
 - 6) Perform the necessary Pre-Health Checks on PROD and STBY side before DR Drill activity.
 - 7) We can use automated tools for DR-Drill activity like SANOFI, Control-M, Epsilon etc.
 - 8) Conduct the RAC (All Node switchover) DR Drill, if you have multiple applications connected to individual node in RAC environment.
 - 9) In a RAC environment, we prefer to perform a single-node switchover by shutting down the other RAC instances, which take longer to become available for the application. Starting the other RAC instances requires additional time.
- To further reduce DR drill time, conducting the RAC DR drill with all DR nodes up and running. This will save time since you only need to open the instances instead of starting them from scratch.

Thanks for taking your time and reading this.

Special Thanks to Mr. Ashish Kumar Singh Sir and my DBA Team.

Please share your opinions, results, suggestions if any.

Join Our WhatsApp DBA community:

<https://chat.whatsapp.com/FK5uiqb22kG6cfsGGX6pbQ>

LinkedIn Profile: [Rahul Kurkute](#)

References:

<https://asanga-pradeep.blogspot.com/2018/03/oracle-data-guard-on-122-cdb-with.html>

<https://eclipsys.ca/dataguard-switchover-in-oracle-cloud-infrastructure-and-oracle-cloud/>

https://docs.oracle.com/en/database/oracle/oracle-database/19/refrn/STANDBY_DB_PRESERVE_STATES.html

<https://www.dbi-services.com/blog/dataguard-oracle-12-2-keeping-physical-standby-sessions-connected-during-role-transition/>

<https://docs.oracle.com/en/database/oracle/oracle-database/19/sbydb/managing-oracle-data-guard-role-transitions.html#GUID-857F6F45-DC1C-4345-BD39-F3BE7D79F1CD>

<https://amitpawardba.blogspot.com/2024/08/step-by-step-oracle-rac-standby.html> (Step by Step Oracle RAC Standby Database Switchover)