CERN

Backup and Recovery

Sébastien Masson

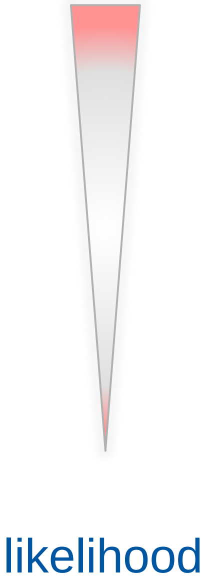
Agenda

- Possible failures and lines of defense
- Backup and Recovery activities in DB group
- Practical cases – Flashback technologies

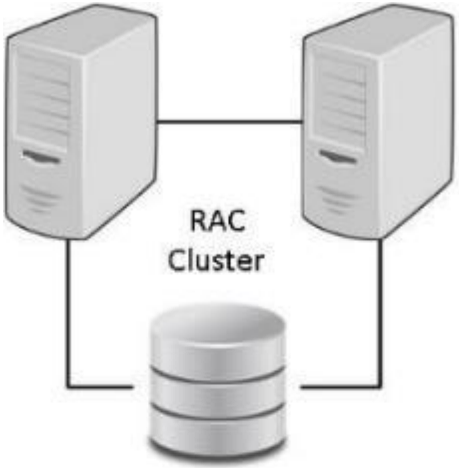
Possible failures and lines of defense

Possible failures

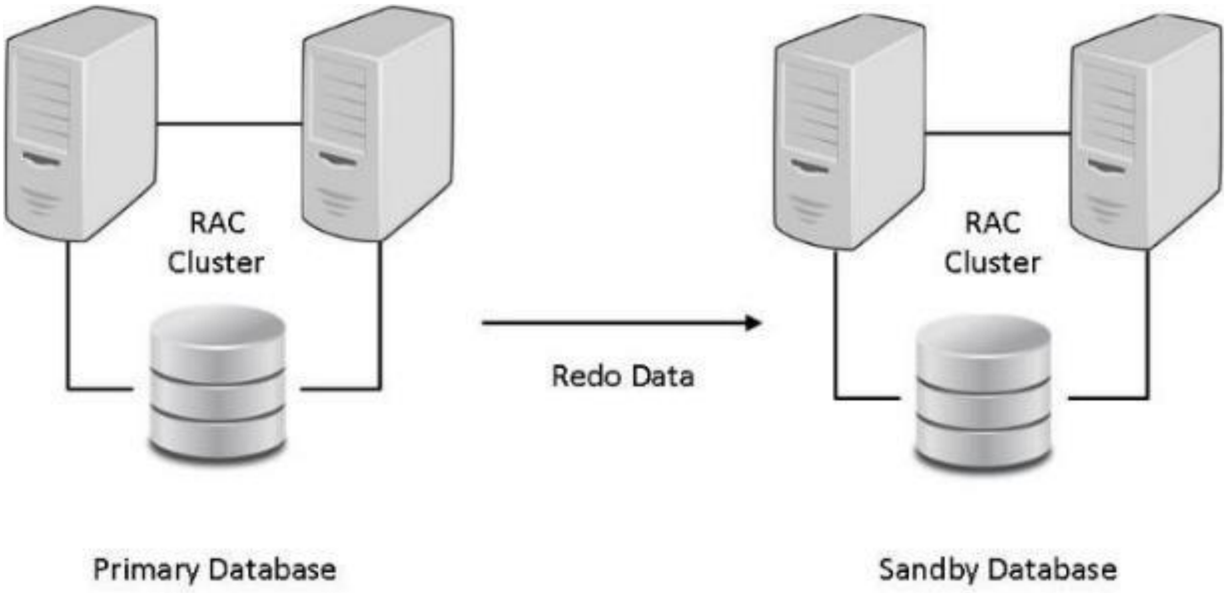
- Human error: mistake with DML(DDL) statement
- Oracle failure at instance level (software)
- Database server failure (hardware)
- Storage failure
- Data corruption
- Disaster



RAC / Data Guard Architecture

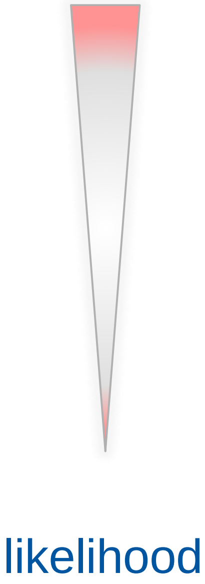


RAC / Data Guard Architecture



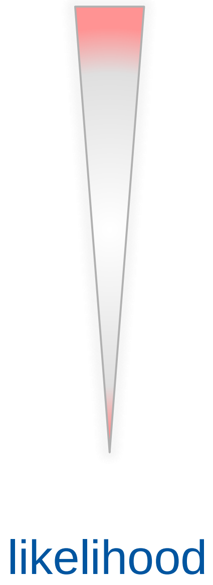
Failures/Solutions

Failure	Solutions	Downtime
Human error	- Flashback technologies - Data Pump exports - Database partial restore	Minutes to Hours of downtime
Oracle Instance	RAC	No downtime most of the time
Database server	RAC	No downtime most of the time
Storage	- Storage itself - Database restore	- No downtime most of the time - Hours to days to restore
Data corruption	- *Oracle Data Guard (detection, repair)* - *Database (detection, repair)* - *Backup (detection)* - Database (partial) restore	- Few hours if small corruption (block recovery) - Hours to days to restore
Disaster	- Oracle Data Guard - Database restore	- Short downtime in case of failover - Hours to days to restore



Failures/Solutions

Failure	Solutions	Downtime
Human error	• Flashback technologies • Data Pump exports • **Database partial restore**	Minutes to Hours of downtime
Oracle Instance	RAC	No downtime most of the time
Database server	RAC	No downtime most of the time
Storage	• Storage itself • **Database restore**	• No downtime most of the time • Hours to days to restore
Data corruption	• *Oracle Data Guard (detection, repair)* • *Database (detection, repair)* • *Backup testing* • **Database (partial) restore**	• Few hours if small corruption (block recovery) • Hours to days to restore
Disaster	• Oracle Data Guard • **Database restore**	• Short downtime in case of failover • Hours to days to restore



Backup and recoveries activities

Backup

- Oracle RMAN (Recovery Manager)
- Any database can be restored to a previous state inside PITR retention period.
- PITR is different on every database (2-3 months usually)
- Complete set of backups on disk, only a subset on tapes (but same retention period) as a secondary protection.
- 4,7 PB needed for all disk backups to satisfy current retention periods.

Backup

- Backups are set up on primary and standby databases:
  - Data files on standby databases
  - Archive logs on primary databases
- Different schedule of backups, depending on the size of the database:
  - Full every week and differential other days
  - Full every two weeks + cumulative every 3 days and differential other days
  - All the archived logs are backed up
- Incremental backups are useful to reduce recovery time.

Recovery

- When needed in case of issue

- Automatic recoveries:
  - Check that our backup strategy is ok:
    - Is everything backed up correctly?
    - Corruption
    - Bugs
  - Weekly or bi-weekly for every database
  - 4 dedicated machines
  - Databases are restored from disk and from tapes

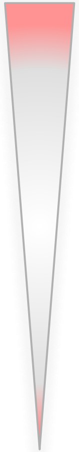
Export

- Oracle Data Pump
- Long retention period
- Different on every database (1 year most of the time)
- Some databases only, with different scopes (schemas, contents…):
  - After a successful automatic recovery
  - Scheduled for few of them
- You can get the data as it was at export time only
- Not a backup and recovery tool, but complementary to RMAN
- Disk and tapes
- Interesting to secure some interventions !

Flashback technologies

Failures/Solutions

Failure	Solutions	Downtime
Human error	• Flashback technologies • Data Pump exports • Database partial restore	Minutes to Hours of downtime
Oracle Instance	RAC	No downtime most of the time
Database server	RAC	No downtime most of the time
Storage	• Storage itself • Database restore	• No downtime most of the time • Hours to days to restore
Data corruption	• *Oracle Data Guard (detection, repair)* • *Database (detection, repair)* • *Backup testing (detection)* • Database (partial) restore	• Few hours if small corruption (block recovery) • Hours to days to restore
Disaster	• Oracle Data Guard • Database restore	• Short downtime in case of failover • Hours to days to restore



Flashback technologies

- Fast method to query or restore objects as they were in the past.
- Past means SCN (System Change Number) or timestamp
- Works only for ‘logical errors’ (related to data manipulation)

- Different technologies:
  - Flashback database, table - restore
  - Flashback queries, version query – query
  - Some other not covered in this presentation

Flashback database (restore)

- ‘Rewind button’ for the **whole, yes, the whole** database.
  - Makes sense with Data Guard on a standby database !
- Faster than classic PITR (which requires restore/recovery)
- You have to contact us before 7 days after the offending SQL statement (the sooner as possible)
- Works thanks to :
  - flashback logs
  - redo logs
- Needs to be enabled, extra-storage is needed
- Some restrictions like tablespace drop,…

Flashback table (restore)

- Reverts back a table to its state in the past
- All of the dependent objects are taken into account (indices, constraints, triggers,..)
- Few restrictions like:
  - Truncate table
  - Modifying or dropping a column (adding is ok)
- Works with Undo data, so no guarantee

```
seb@BRTEST:SQL> create table test (id number(10))
tablespace DATA01;
Table created.

seb@BRTEST:SQL> alter table test enable row movement;
Table altered.

seb@BRTEST:SQL> select sysdate from dual;
SYSDATE
--------------------
12-NOV-2017 16:18:34

seb@BRTEST:SQL> insert into test values (1);
seb@BRTEST:SQL> insert into test values (2);
seb@BRTEST:SQL> insert into test values (3);
seb@BRTEST:SQL> commit;

Commit complete.

seb@BRTEST:SQL> select count(*) from test;

COUNT(*)
--------------
 3
```

```
seb@BRTEST:SQL> FLASHBACK TABLE test TO TIMESTAMP
TO_TIMESTAMP('2017-11-12 16:18:34', 'YYYY-MM-DD
HH24:MI:SS');

Flashback complete.

seb@BRTEST:SQL> select count(*) from test;

COUNT(*)
--------------
 0
```

Flashback drop (restore)

- Cancel a ‘drop table’ statement (if done without purge option ☺)
- All of the dependent objects are taken into account (indexes, constraints, triggers,..)
- Works if recycle bin is enabled (enabled by default)
- Objects in recycle bin can be purged manually
- Objects can disappear from recycle bin if there is a space pressure in the object’s tablespace → no guarantee

```
seb@BRTEST:SQL> drop table test;
Table dropped.

sys@BRTEST:SQL> select owner,OBJECT_NAME,ORIGINAL_NAME
size_mo from dba_recyclebin;

OWNER         OBJECT_NAME                              ORIGINAL_NAME
----------        --------------------------------------------------          --------------------
SEB              BIN$XjBXUkJ4hbngU/kGEApZAQ==$0      TEST

1 row selected.

sys@BRTEST:SQL> select tablespace_name from dba_segments where
segment_name='BIN$XjBXUkJ5hbngU/kGEApZAQ==$0';

TABLESPACE_NAME
------------------------------
DATA01
```

```
seb@BRTEST:SQL> flashback table test to before drop;
Flashback complete.

seb@BRTEST:SQL> desc test;

Name                Null?    Type
 ------------------- ----------- ----------------------------------------
 ID                            NUMBER(10)
```

Flashback query (query)

- Flashback query: See the contents of a table in the past
- SCN or timestamp
- Rely on undo data
- Some limitations too

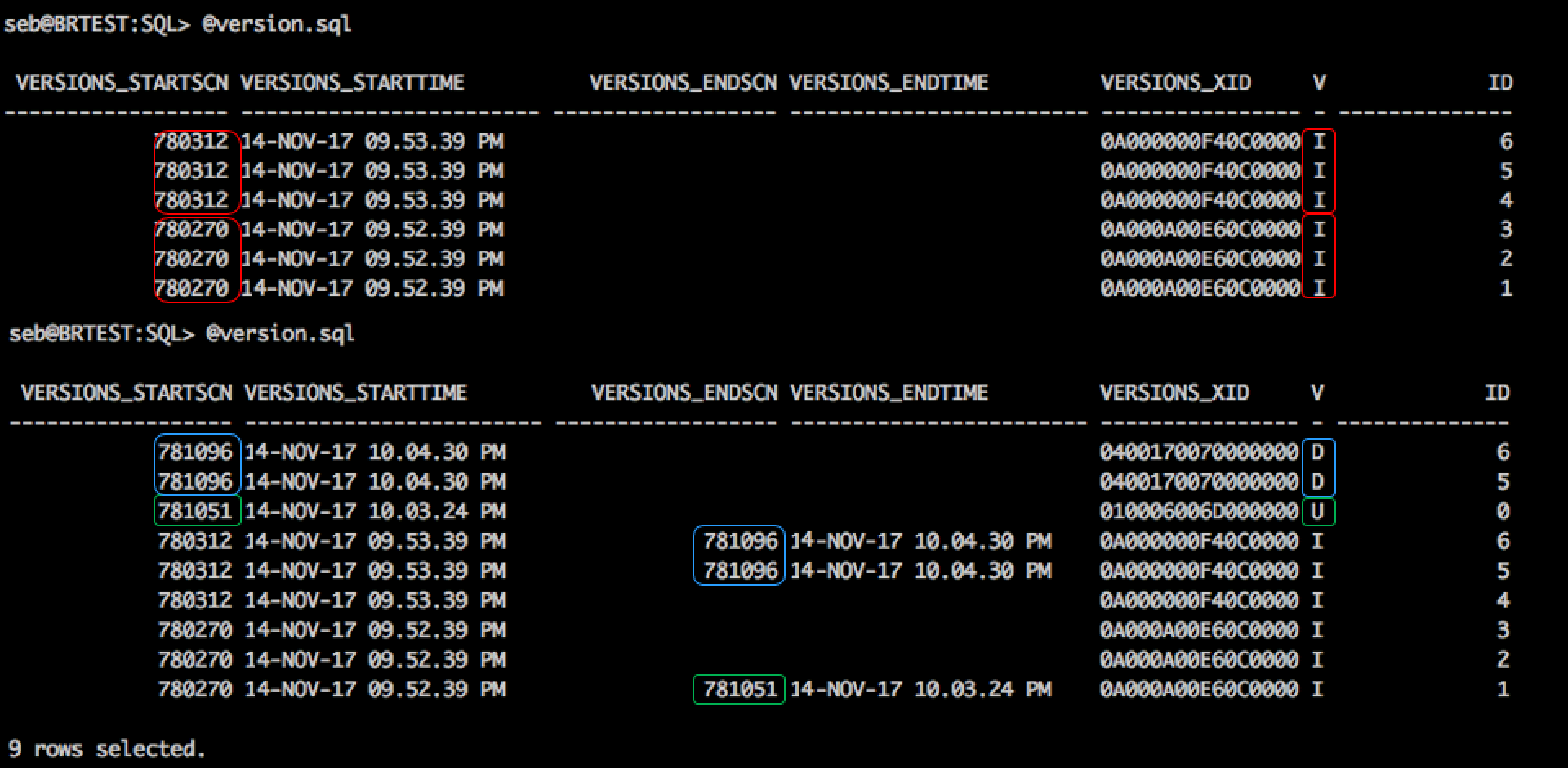
```
seb@BRTEST:SQL> create table test (id number (10));
seb@BRTEST:SQL> insert into test values (1);
seb@BRTEST:SQL> insert into test values (2);

seb@BRTEST:SQL> select current_scn from v$database;

  CURRENT_SCN
--------------
                  760311

seb@BRTEST:SQL> commit;

Commit complete.

seb@BRTEST:SQL> select current_scn from v$database;

  CURRENT_SCN
--------------
                  760338
```

```
seb@BRTEST:SQL> select * from  test as of scn
760311;

no rows selected



seb@BRTEST:SQL> select * from  test as of scn
760338;

                    ID
--------------
                     1
                     2
```

Flashback Versions Query (query)

- See the different versions of rows in a time interval
- You can track the changes
- Works with undo data

```
SELECT
        versions_startscn, versions_starttime,
        versions_endscn, versions_endtime,
        versions_xid, versions_operation,
        table_columns
FROM   table
        VERSIONS BETWEEN TIMESTAMP
TO_TIMESTAMP(begin_interval)
        AND TO_TIMESTAMP(end interval)
(WHERE …);
```



```
seb@BRTEST:SQL> @version.sql

 VERSIONS_STARTSCN VERSIONS_STARTTIME          VERSIONS_ENDSCN VERSIONS_ENDTIME          VERSIONS_XID     V         ID
 ----------------- ------------------------- ----------------- ------------------------- ---------------- - ----------
            780312 14-NOV-17 09.53.39 PM                                                  0A000000F40C0000 I          6
            780312 14-NOV-17 09.53.39 PM                                                  0A000000F40C0000 I          5
            780312 14-NOV-17 09.53.39 PM                                                  0A000000F40C0000 I          4
            780270 14-NOV-17 09.52.39 PM                                                  0A000A00E60C0000 I          3
            780270 14-NOV-17 09.52.39 PM                                                  0A000A00E60C0000 I          2
            780270 14-NOV-17 09.52.39 PM                                                  0A000A00E60C0000 I          1

seb@BRTEST:SQL> @version.sql

 VERSIONS_STARTSCN VERSIONS_STARTTIME          VERSIONS_ENDSCN VERSIONS_ENDTIME          VERSIONS_XID     V         ID
 ----------------- ------------------------- ----------------- ------------------------- ---------------- - ----------
            781096 14-NOV-17 10.04.30 PM                                                  0400170070000000 D          6
            781096 14-NOV-17 10.04.30 PM                                                  0400170070000000 D          5
            781051 14-NOV-17 10.03.24 PM                                                  010006006D000000 U          0
            780312 14-NOV-17 09.53.39 PM                781096 14-NOV-17 10.04.30 PM     0A000000F40C0000 I          6
            780312 14-NOV-17 09.53.39 PM                781096 14-NOV-17 10.04.30 PM     0A000000F40C0000 I          5
            780312 14-NOV-17 09.53.39 PM                                                  0A000000F40C0000 I          4
            780270 14-NOV-17 09.52.39 PM                                                  0A000A00E60C0000 I          3
            780270 14-NOV-17 09.52.39 PM                                                  0A000A00E60C0000 I          2
            780270 14-NOV-17 09.52.39 PM                781051 14-NOV-17 10.03.24 PM     0A000A00E60C0000 I          1

9 rows selected.
```

To conclude…

Different ways to recover data

Solution	Time needed	Pros	Cons
Flashback technologies	*	• Fast • Easy to use	• Works on a limited period of time • Restrictions • Data Guard ‘needed’ on database scope
Data pump	**	• Quite fast • Very flexible	• Need to have an usable data pump export
Partial database restore	******	• Always works • Can be far quicker than the full restore	• Takes a lot of time
Full database restore	*************	• Always works • Fully scripted (automatic recoveries)	• Takes a lot of time