

THE ADVENTURE OF UPGRADING TO 11.2

whoami

- 12 years experience with Oracle Database
- Oracle Database & Oracle RAC fanatic
- Oracle Certified Master
 - OCP DBA 9i/10g
 - OCE RAC
 - OCE SQL
- DBA manager at Skrill
- <http://blog.yavor.info>

About this presentation

- I'm not going to tell you how to upgrade
 - The documentation is quite good at this
 - MOS is handy also
- I will share some experience from the real world
 - Things not written in the documentation
 - Or not emphasized
 - Things not supposed to be written in the documentation

Agenda

- Task definition
- Preparation
- Tests
- The upgrade
- Our findings

Task definition

- Critical 24x7 database
- Logical and multiple Physical standbys (Sync and Async)
- PCI compliance
 - We cannot take real database out of the PCI compliant environment
- The old system:
 - 10.2.0.4.1 Database
 - 11.1.0.7. ASM
- Target: 11.2.0.2.3
- No RAC

Preparation: planning

- Hardware
 - Most tests run best with production-like servers and storage
 - The more servers and storage you have, the more tests you can do in parallel
- People
 - DBAs, SAs, QAs, Developers
- Agreements with the business
 - Downtime
 - The point of no return
 - The upgrade shouldn't be combined with other things, like application releases, hardware maintenance, etc.
 - Some days will be needed after the upgrade for fixing post-upgrade issues
- Prepare production environment
 - Downtime is needed for ASM upgrades

Preparation: choice of upgrade strategy

- DBUA
 - Takes care of lot of things
 - Easier
- Manual
 - More control
 - Repeatable results
- Prepare and test upgrade plan before all other tests
 - Every time you need a test database, use the same upgrade “script” as in the plan

Preparation: check for bugs

- We found ~150 bugs for 11.2.0.2.3 database and ~40 for ASM that may hit us
- We had to choose wisely
 - Data corruption
 - Instance crash
 - Wrong results

Functional tests

- Prepare primary, logical and physical standbys
- **2 full-size environments: 10g and 11g**
- Real production data

- 5 QAs, 1 DBA, sysadmin
- 2 weeks

- Check database parameters
- Block all outgoing traffic from test servers (email, db links, etc)
- Do not overload the database – e.g. do not start all toughest reports in the same time

Automatic tests

- Prepare primary, logical and physical standbys
- **2 full-size environments: 10g and 11g**
- Production data

- 2 AQAs, 1 DBA, sysadmin
- 2 weeks

- Check database parameters
- Check APP servers
- Do not overload the database – try to execute “normal” load

Real application testing – capture details

- Capture did not add noticeable load to the database
- Oracle's advise: restart source database before capture
 - Nonsense for busy production system
- Make sure you have enough space for capture files
 - Average size: `2 * Bytes received via SQL*Net from client`
- After capture, export AWR using
`DBMS_WORKLOAD_CAPTURE.EXPORT_AWR`

Real application testing – replay details

- Restore database copy up to the SCN of capture start
`select name, start_scn, status from dba_workload_captures;`
- Upgrade/patch database copy
- Check database parameters!
- Pre-process captured data – **takes time!**
(note 560977.1)

`DBMS_WORKLOAD_REPLAY.PROCESS_CAPTURE`

- Calibrate the replay (runs in a few seconds)

`wrc mode=calibrate replaydir=./replay`

- Run the replay – takes more time than capture – up to 2x
(this is for sync replays)
- Import AWR
`DBMS_WORKLOAD_CAPTURE.IMPORT_AWR`

Real application testing – getting results

- **Basic “feedback”**

`DBMS_WORKLOAD_REPLAY.REPORT`

- **More data**

`dbms_workload_replay.compare_period_report`

- **Most data**

`dbms_workload_repository.awr_diff_report_html`

Real application testing - jobs

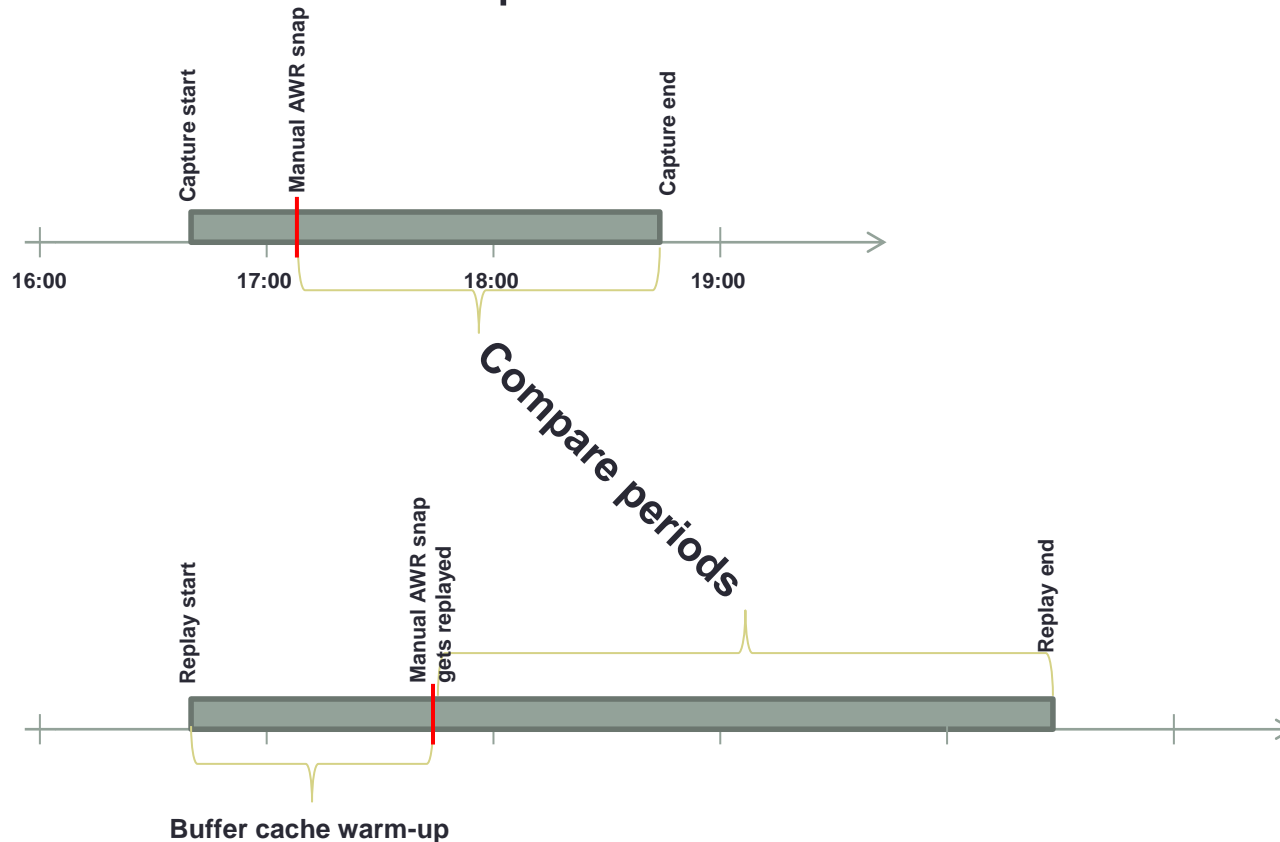
- Jobs are not captured
- JOB_QUEUE_PROCESSES should be set to normal value
- Oracle advise: set server clock to the time of capture

Real application testing - period

- Not for long periods
 - Too much data
 - Slow pre-processing
 - Hard to analyze
- Not for short periods
 - Cold buffer cache
- Pre-processing of captured data takes time
- Replay (sync) takes more time than capture (up to 2x)
 - Look at DB time, not wall clock time

Real application testing – buffer cache warm-up

- Longer period to mitigate the difference
- Manual AWR snapshot



DBA tests

- Backup / Restore
- Data guard (switchover/failover)
- ASM
- Load tests (host, IO, network)

Pre-upgrade

- Upgrade ASM if you have downtime windows
- Install software, patches
- Run pre-upgrade checks. Fix most things
 - Purge recycle bin
 - Drop orphan distributed transactions
 - Gather dictionary stats
 - etc.
- Prepare spfiles, password files, etc
- Write **detailed step-by-step** upgrade plan
- Physical standbys instead of backups
 - On production-like storage 😊

The UPGRADE

- We've done it early in the morning (started 4:00)
- 2 teams of 2 DBAs
 - One team to upgrade primary database
 - One team to upgrade logical standby
- 1 DBA to do the synchronization with other teams (sysadmins, QAs, managers)
- Some DBAs should be sent home shortly after the upgrade, to take some rest and prepare for “night” shift

Findings: upgrade with standbys

- For the upgrade part
 - Physical standbys should be started from 11.2 home and the upgrade gets propagated automatically. Works like charm
 - Logical standby have to be upgraded separately, as if it is a primary database. We have done the upgrades in parallel to save time
- COMPATIBLE parameter
 - Have to be set to 11.2 on standbys, then on primary
 - If you have cascade standbys, start from the last

http://download.oracle.com/docs/cd/E11882_01/server.112/e25608/upgrades.htm#CIHJIICA

Findings: auditing

- In 11g auditing “by session” is deprecated
- All auditing is done “by access”
- We had to rethink our auditing strategy

<http://blog.yavor.info/?p=650>

Findings: direct path read for the serial FTS

- New feature – full table scans of large tables may use direct path read and skip the buffer cache
- For our system, all tests showed this leads to poor performance
- We have disabled this by setting event 10949
- We plan to log an SR for this
- <http://dioncho.wordpress.com/2009/07/21/disabling-direct-path-read-for-the-serial-full-table-scan-11g/>

Findings: lob writes

- All tests showed huge increase in “lob writes” waits stat (like 20x)
- The lobs are handled differently in 11g
- We could not find the root case
- There is no performance impact in our system
- Probably the lob writes are better instrumented on 11.2?

http://www.orainternals.com/papers/oracle_11g_performance_features_riyaj_doc.pdf

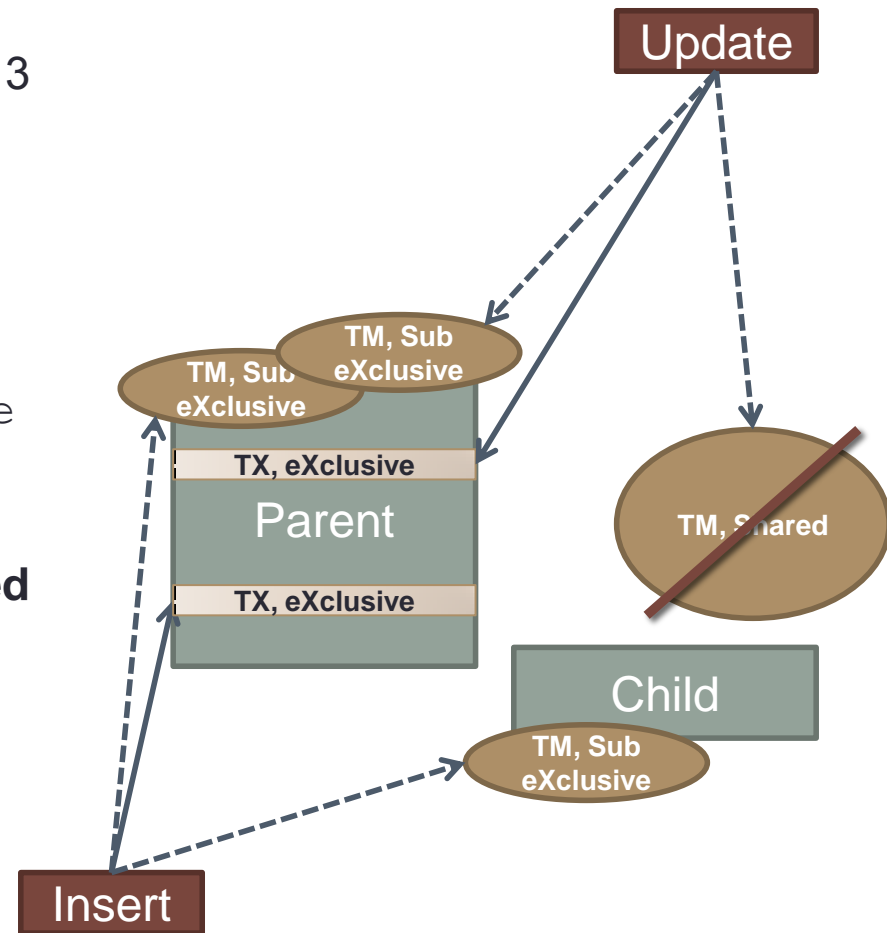
Findings: resource manager

- In 11g resource manager is enabled by default
- Resource plans are switched on scheduler window change
- The windows are fine for 8x5, but not for 24x7

<http://oracledoug.com/serendipity/index.php?/archives/1604-Resource-Manager-and-11g.html>

Findings: TM locks

- In 11g, changes in parent table's data put level 3 (SX - Sub-eXclusive) TM locks on child tables
 - instead of 10g's level 2 (SS - Sub-Share)
 - The change in lock modes behavior introduced intentionally by fix 5909305 in 11.1 onward
- SX lock is incompatible with S and SSX (Share and Share Sub-eXclusive)
- Update/Delete on parent requires S lock on child, **if the foreign key column is not indexed**
- The result – INSERT in parent blocks all other sessions from doing UPDATE or DELETE on parent
 - Even if there are NO rows affected on child table!



Post-upgrade

- Only a handful of queries with broken execution plans
 - Most fixed by Tuning advisor
- Some new access paths and optimizations, some queries are performing better
 - Nothing exciting out of the box
- Logical standby performs much better
 - Lag is way shorter, apply is faster
- Automatic tuning advisor
 - Looks useful, we are still considering if we want to use it
- Better monitoring in GridControl

Interesting: DWH and dynamic sampling

- The following query works in <0.01 sec:

```
SELECT * FROM fact_table WHERE fact_id = 123;
```

- The following query runs for 65 minutes

```
SELECT * FROM fact_table WHERE fact_id in (123, 456, 789);
```

- SQL trace showed where the time is lost – the DB is executing the following:

```
SELECT /* OPT_DYN_SAMP */ /*+ ALL_ROWS IGNORE_WHERE_CLAUSE  
NO_PARALLEL(SAMPLESUB) opt_param('parallel_execution_enabled', 'false')  
NO_PARALLEL_INDEX(SAMPLESUB) NO_SQL_TUNE */ NVL(SUM(C1),:"SYS_B_000"),  
NVL(SUM(C2),:"SYS_B_001")  
FROM  
(SELECT /*+ IGNORE_WHERE_CLAUSE */ : "SYS_B_002" AS C1, CASE WHEN...
```

- Workaround: setting
OPTIMIZER_DYNAMIC_SAMPLING=0 on session level
- SR on the way...

Major benefits for us

- Premium support
- Active DataGuard
- Result cache
- Logical standby works on compressed segments (except for HCC)
- Snapshot standby

For the next upgrade

- Next time we will try to do “zero downtime” upgrade

Q & A

<http://blog.yavor.info>

yavor [at] yavor [dot] info