PayPal™

Internals of Active Dataguard

Saibabu Devabhaktuni

PayPal DB Engineering team

- Sehmuz Bayhan – Our visionary director
- Saibabu Devabhaktuni – Sr manager of DB engineering team
  http://sai-oracle.blogspot.com (blog)
- Kyle Towle – It can’t get any better, expert at DB and Systems architecture, scalability, and availability.
- Amit Das – One of the very few RAC/Exadata experts in the world
- Dong Wang – Worlds best Goldengate expert
- John Kanagaraj – Coauthored "Oracle Database 10g : Insider Solutions“ and Oracle ACE.
- Sarah Brydon – One of the very few Oracle Certified Masters.

Scope

- This presentation is applicable for Oracle 11.2.0.2
- Some of the observations made here may be incorrect, please verify it in your environment prior to using any recommendations.

Agenda

- Redo apply mechanism
- Read only queries
- State of the standby
- Testing applications for Active Dataguard (ADG)

Redo apply on primary

- Read data block buffers in current mode
- Generate redo for intended change in PGA
- Redo generated for undo and data blocks
- Copy redo from PGA to redo buffer
- Check block for SCN and seq prior to redo apply
- Apply redo to undo and data blocks
- Redo record contains change vectors for undo and data blocks

Redo apply on standby

- MRP process read redo typically in chunks of 1MB
- MRP process sort the redo in SCN order
- Assign redo to be applied to recovery slaves (probably by hash of data blocks)
- Recovery slave use asynch I/O calls up to 32 blocks in a batch
- Total outstanding asynch I/O calls per recovery slave can be up to 1024 in 11g (used to be 4K in 10g)
- Keep applying redo to the blocks available in SGA
- Rely on 2nd pass redo apply for blocks not in SGA

Redo shipping

- Redo shipping from primary is done in streaming mode for LGWR SYNC/ASYNC
- Check “Redo KB read*” values in v$sysstat
- Redo to be shipped can be compressed
- Query v$standby_log for status of real time redo shipping
- Make sure to create standby logfiles on primary
- Watch out for case sensitive password in orapwd file across the primary and standby

Redo apply metrics

- Query v$managed_standby for MRP progress
- Current_scn, standby_max_data_delay and v$recovery_progress rely on same underlying memory structure
- Check for recovery read and checkpoint complete wait events in ash and v$event_histogram
- Check for apply time and checkpoint time per log in v$recovery_progress
- Query current_scn in v$database or v$dataguard_stats for standby lag [1]

Checkpoint on standby

- Mandatory checkpoint at every log boundary
- Recovery slaves wait on log boundary checkpoint
- No redo apply during log boundary checkpoint
- Incremental recovery checkpoints keep log boundary checkpoint duration smaller
- Checkpoint performed prior to stopping MRP
- Parameters “_defer_log_boundary_ckpt” and “_defer_log_count” define the behavior (not fully implemented in 11g)
- Recovery until consistent required after crash of ADG

Redo apply on RAC

- MRP process read redo in chunks from all threads
- MRP process merge all the redo and sort it
- Only one instance apply the redo at a time
- MRP may wait if we redo is not arrived for any thread
- MRP process reading redo can’t be parallelized
- Log boundary checkpoint performed for each thread

Redo apply contd.

- New datafile creation on standby is done by single recovery slave while rest of the slaves wait for it
- Redo apply stop if new datafile creation fails
- Failed datafile creation can be recreated manually while ADG is still open and MRP stopped
- MRP acquires “MR” type lock on all online datafiles
- Offline or online of a datafile require MRP to stop
- Partial standby is supported but not integrated with MRP

More on redo apply

- No redo on primary for commit time block cleanouts unless “_log_committime_block_cleanout” is set
- Nologging on primary causes effected blocks to be corrupted on standby
- Corrupted datafiles on standby can be detected by running dbv or rman
- Flashback on standby works just like on primary
- Apply patch 10094823 on 11.2.0.2 and below prior to using block change tracking on the standby

Optimizing redo apply

- Use parallel redo apply (default in 11g and beyond)
- Set parallel_execution_message_size to 65536
- Use fewer DB writer processes and parallel recovery processes if I/O latency is a true bottleneck
- Match SGA size to primary if possible
- Keep redo log file size as big as possible but make sure to meet recovery point objectives on primary
- Use same redo log file size across all RAC nodes
- Use real time apply (be aware of bug 5956646, RTA can leave standby in unrecoverable state) [1]

Reads on ADG

- 50% of buffer cache reserved for reads in ADG (as defined by “_recovery_percentage”)
- Row cache structures built on standby like primary
- DDL invalidations to happen like on primary
- Current_scn in v$database used for query SCN
- Buffers in recovery state (5) can be used for reads
- Buffer clones happen like on primary (state 3 or state 2, shared current, in RAC used for consistent reads)

Role of Undo

- Undo is used for consistent reads like on primary
- Redo for data and corresponding undo applied separately
- Undo block not in SGA could delay redo apply
- Query reading data block buffer wait for corresponding undo block to be current with the SCN
- No select for update operations permitted as it requires changes to transaction table and data blocks

Commit time block cleanout

- Query process perform block cleanout for each execution
- New buffer clone for each execution
- No block cleanout in current or recovery state buffers
- Blocks cleaned out on standby never written to disk
- Ora_rowscn will be different for each execution
- Demo

Read only tables

- Cleanouts happen just like for read write tables
- Since table is already made read only on primary, cleanouts on standby continue to happen for each query execution forever
- Same thing can happen for objects under read only tablespaces
- Parallel queries also incur same overhead and they typically perform block cleanouts in PGA
- Work around is to perform full cleanout on primary prior to making table or tablespace read only

More on reads

- DDL invalidation issue for synonyms on standby (demo)
- Set “_log_committime_block_cleanout” to TRUE on primary if possible, redo overhead is less than 5%.
- Any queries requiring recompilation of objects like views, procedures will fail.
- Any user login which require update to user$ will fail (i.e. password expiry time, etc)

Read performance metrics

- V$ views populated like on primary
- Setup standby statspack to persist data from v$sql, ASH, and other views
- Oracle can’t persist query column predicate information to col_usage$ and it may effect query plans based on how stats are gathered
- Not possible to set system statistics like CPU or I/O speed on ADG
- Some of the enhancements to sql plan related features which require data to be persisted will not work on ADG

Standby state

- By design, not all control file changes make it to the standby
- Supplemental logging can be different on standby
- Supplemental log level can be different on standby
- Goldengate and logical standby may not work properly post switchover
- DG switchover doesn’t report these control file changes
- Make it part of pre switchover checklist

Standby state

- Force logging can be different on standby
- Batch jobs may perform nologging activity and hence it can invalidate former primary post DG switchover
- Flashback state can be different on standby
- Flashback state can be different at tablespace level
- Flashback files should not be copied over, they are valid for a given database only
- Fix is to run same alter database commands on the standby or refresh standby control file from primary

Standby state

- Control file contents can be significantly different on standby, especially if RMAN is used [1]
- Don’t rely on unrecoverable columns in v$datafile (as they are not updated by design)
- Temp files can be different on standby
- More temp files can be created on standby
- GG extract will not work on ADG with ASM
- Redo apply instance crash triggers other RAC nodes to shutdown

Lost writes

- Lost writes can happen on primary and standby
- Recommended to set DB_LOST_WRITE_PROTECT
- Not all lost writes can be detected by above parameter
- DB_LOST_WRITE_PROTECT relies on the assumption effected blocks will be eventually read
- Most of the lost writes happen outside of Oracle code
- No easy way to check for lost writes

Manual checks for lost writes

- Blocks across primary and standby can’t be binary compared even if they are at same SCN
- Data can be compared at same SCN
- Use sum of ora_hash on all records at block or segment level
- Ora_rowscn can’t be used for comparison
- Indexes can be checked by computing hash of all key columns using index fast full scan and repeat it with full table scan at same SCN
- Run rman backup validate check logical for non lost write corruption

Testing application for ADG

- Test application for benefits of ADG prior to purchasing ADG license option
- Query percentage of read only queries versus read write in ASH by XID column
- Dba_hist_active_sess_history can also be queried
- Select for update queries need to be flagged
- Check MOS note 1206774.1 for event 3177
- Event 3177 only log violations in trace file
- spare statistic 1 in v$sysstat also report violations

QA testing without ADG

- QA testing for ADG doesn’t have to use ADG in QA
- Create separate user (if possible) for read only flow
- Create a dual like dummy table with primary key
- Insert one row in dummy table and make it read only
- Create view joining dummy table with live table
- Create synonym for view if necessary
- All DML’s will fail as required for ADG test
- Demo

Enhancing ADG

- ER, 13719619, for ability to test read only apps
- ER, 12781212, for parallelizing datafile creation
- Parameter “_log_committime_block_cleanout” set to TRUE by default to reduce number of cleanouts
- No mandatory log boundary checkpoint
- Integrate MRP for partial standbys (needed on RAC)
- ER, 12802025, for RMAN to check datafiles across primary and standby (for lost write detection)

Summary

- ADG is one of the best Oracle products
- Redo apply keeps getting better
- Better read traffic option than Goldengate
- More block cleanout's on ADG, but impact is low
- Lost write detection needs to be supplemented
- Key database properties (i.e. nologging, supplemental logging, etc) can be different on ADG
- Partial standby possible for consolidated primary databases