Oracle Database Smart Flash Cache Overview

Harald van Breederode
Oracle University
28-NOV-2011

About Me

- `#include <std/disclaimer.h>`
- Senior Principal DBA Trainer – Oracle University
- 25 years Unix Experience
- 12 years Oracle DBA Experience
- Oracle8*i*, 9*i*, 10*g* and 11*g* OCP
- Oracle10*g* and Oracle11*g* OCM
- DBA Certification Exam Team Reviewer
- DBA Curriculum Development Reviewer
- Blog: prutser.wordpress.com
- Visually Impaired (Legally Blind)

ORACLE
Certified Master
Oracle Database 11*g*
Administrator

ORACLE

Agenda

- Overview
- Buffer Cache basics
- Oracle Database I/O basics
- Oracle Database and Flash technology
- Database Smart Flash Cache architecture
- Database Smart Cache parameters
- Database Smart Flash Cache configurations
- Database Smart Flash Cache instrumentation
- Demo
- References
- Questions & Answers

Overview

The goal of this presentation is to answer the following questions:

- How to use flash products in Oracle Database environments
- What is the Oracle Database Smart Flash Cache
- How to setup the Oracle Database Smart Flash Cache

Buffer Cache Basics

The main buffer cache structures are:

- Buffers
  - Each buffer may hold one block image at any given time
- Buffer headers
  - Store metadata about contents of the buffers
  - Act as buffer cache management structures
- Buffer pools
  - Collections of buffers used for the same purpose
  - Enable multiple block size support
  - Enable multiple DBWR's
  - Different management algorithms

Buffer Cache Management

Cache is managed by doubly linked lists:

- REPL
  - Buffers containing block images being used
- REPL-AUX
  - Buffers ready to be used for I/O or CR build
- WRITE
  - Dirty Buffers requiring I/O
- WRITE-AUX
  - Dirty Buffers with I/O in progress

Note: AUX lists avoid wasteful scanning

Buffer Cache I/O

- Server processes look for an available buffer on REPL-AUX
  - Buffer gets moved from REPL-AUX to REPL
  - Insertion point dictated by pool algorithm
  - Data block is read in buffer
- Servers move dirty buffers to WRITE during free buffer search
- DBWn writes dirty buffer contents to database
  - Buffer gets moved from WRITE to WRITE-AUX
  - Buffer is moved back to REPL-AUX Once block is written
- DBWn writes upon request
  - Make free buffers
  - Checkpoint

Basic Oracle Database I/O

A server process needs to access an Oracle database block:

- If block resides in the Oracle Buffer Cache
  - Logical read
  - Usually takes a few microseconds to complete
- If block does not resides in the Oracle Buffer Cache
  - Physical read
  - Oracle makes request to underlying layer (probably disk)
  - Usually takes several (5-15) milliseconds to complete
  - Response time depends on:
    - Rotating speed => Rotational delay
    - Type of disk: SATA, SAS, SAS2
    - Architecture: JBOD, SAN, NAS
    - Concurrency and amount of I/O

Improving Database I/O Performance

- Application tuning
  - Is this supported by application vendor?
- SQL statement tuning
  - Is this allowed by application vendor?
- Database tuning
  - Is this allowed by application vendor?
- Faster or more CPUs
  - Does it affect license costs?
- Faster or more memory
  - Are there enough memory slots available?
- Faster or more disks
  - Are there free drive bay's available?

About Flash Technology

- Fast!
- But, quite an investment
- SSD disks
- Flash cards
  - Sun Flash Accelerator F20
  - Fusion-io ioDrive
- Flash based storage array's
  - Sun Storage F5100 Flash Array

A Few Examples

- Sun Flash Accelerator F20
  - PCI-Express card
  - 96GByte capacity presented as 4 disk drives
  - Over 100,000 random IOPS
- Sun Storage F5100 Flash Array
  - 1RU form factor (rack mountable)
  - Up to 1.92TB capacity
  - Over 1 million IOPS
- Fusion-io ioDrive
  - PCI-Express card
  - 160, 320 or 640 Gbyte capacity presented as 1 disk drive
  - Over 100,000 random IOPS

Where Do We Put Them?

- Inside the SAN or NAS
  - Data has to travel over slow I/O channels
  - Flash reliability might be an issue
- Inside the database server
  - Sharing database files for RAC becomes impossible
  - Data is close to the database instance

What Should We Store On Flash

- Online redo log files?
- Control files?
- Data files?
- Temp files?
- Most frequently used data?

Introducing the Database Smart Flash Cache

- Not to be confused with Exadata Smart Flash Cache
- Unique Oracle DB feature on Oracle Linux and Oracle Solaris
- Expanding the buffer cache on flash storage
  - L1 cache - main memory
  - L2 cache - flash cache
- L2 buffer headers reside in L1 cache 100 bytes/buffer
- L2 buffers reside on flash storage
- 3 new doubly linked lists
  - L2REPL - Blocks that are cached in Smart Flash Cache
  - L2KEEP - Blocks to be kept in Smart Flash Cache
  - L2WRITE - Blocks ready to be written to Smart Flash Cache
- L2KEEP has priority over L2REPL

Buffer Cache Management Revisited

- Cold clean buffers aging out are moved to L2WRITE unless
  - Block is already in the Smart Flash Cache
  - Object is marked with `FLASH_CACHE` set to `NONE`
  - These are directly reused or moved to REPL-AUX
- DBWR writes blocks from L2WRITE into Smart Flash Cache
- Buffers are either placed on L2REPL or L2KEEP based on `FLASH_CACHE` attribute
  - `DEFAULT` => L2REPL
  - `KEEP` => L2KEEP
- Buffer is moved from L2WRITE to REPL-AUX
- If DBWR is busy writing to disk:
  - DBWR does not write to Smart Flash Cache
  - Buffers are directly moved to REPL-AUX

Basic Oracle Database I/O Revisited

A server process needs to access an Oracle database block:

- If block resides in the Oracle Buffer Cache
  - Logical read
  - Usually takes a few microseconds to complete
- If block resides in Database Smart Flash Cache
  - Optimized physical read
  - Usually takes several (1-5) tents of a millisecond to complete
- If block does not resides in the Oracle Buffer Cache
  - Physical read
  - Oracle makes request to underlying layer (probably disk)
  - Usually takes several (5-15) milliseconds to complete

Database Smart Flash Cache Parameters

- DB_SMART_FLASH_CACHE_FILE
  - Location of Smart Flash Cache
  - 1 O/S or ASM filename
  - File cannot be shared by multiple databases or instances
- DB_SMART_FLASH_CACHE_SIZE
  - Size of the Database Smart Flash Cache
  - Guideline: 2 to 10 times db_cache_size
  - No dynamic resize
  - Set to 0 to disable Smart Flash Cache
  - Set to original size to re-enable it

What Will Be Cached?

- Caching is controlled by `FLASH_CACHE` object attribute
  - `NONE` – No caching
  - `DEFAULT` – Normal caching
  - `KEEP` – Cache as long as possible
- Visible in:
  - `DBA_CLUSTERS`
  - `DBA_INDEXES`
  - `DBA_IND_[SUB]PARTITIONS`
  - `DBA_LOB_[SUB]PARTITIONS`
  - `DBA_OBJECT_TABLES`
  - `DBA_SEGMENTS`
  - `DBA_TABLES`
  - `DBA_TAB_[SUB]PARTITIONS`

Example Setups

- Using Sun Flash Accelerator F20
  - Create ASM disk group on all 4 flash disks
  - Set `DB_SMART_FLASH_CACHE_FILE` to +DG
  - Set `DB_SMART_FLASH_CACHE_SIZE` to desired cache size
  - Multiple databases or instances can share ASM disk group
- Using Fusion-io ioDrive
  - Create one or more Linux partitions or Solaris slices
  - Set `DB_SMART_FLASH_CACHE_FILE` to partition or slice
  - Set `DB_SMART_FLASH_CACHE_SIZE` to partition/slice size
- Alternative setup
  - Create Linux or Solaris filesystem on flash drive(s)
  - Set `DB_SMART_FLASH_CACHE_FILE` to filesystem file
  - Set `DB_SMART_FLASH_CACHE_SIZE` to desired cache size

When to configure Database Smart Flash Cache?

- Your database is running on Oracle Solaris or Oracle Linux
- AWR indicates that a larger buffer cache is beneficial
- DB file sequential read is a top wait event
- You have spare CPU capacity
- In case of a RAC database each instance must have its own DB Smart Flash Cache

Database Smart Flash Cache Instrumentation

- Statistics
  - flash cache eviction: aged out
  - flash cache eviction: buffer pinned
  - flash cache eviction: invalidated
  - flash cache insert skip: DBWR overloaded
  - flash cache insert skip: corrupt
  - flash cache insert skip: exists
  - flash cache insert skip: modification
  - flash cache insert skip: not current
  - flash cache insert skip: not useful
  - flash cache inserts
  - physical read flash cache hits

Database Smart Flash Cache Instrumentation

- Wait events
  - `db flash cache dynamic disabling wait`
  - `db flash cache invalidate wait`
  - `db flash cache multiblock physical read`
  - `db flash cache single block physical read`
  - `db flash cache write`
  - `write complete waits: flash cache`

Case Study

- MCX Administration Services BV – The Netherlands
  - E-Business Suite management and hosting provider
- The problem:
  - Inconsistent database performance at peak hours
  - Blade servers at maximum RAM capacity
  - NAS storage reached its maximum performance
- The solution:
  - Database Smart Flash Cache using Fusion-io ioDrive
  - OLTP and batch performance increased 60-80%
  - Consistent performance at all times
  - No new database servers and/or storage expansion!
- Reference:
  - http://fusionio.biz/case-studies/mcx/

Live Demo!

References

- Oracle Database Administrators Guide
  - http://docs.oracle.com/cd/E11882_01/server.112/e25494/toc.htm
- Database Smart Flash Cache Whitepaper
  - http://www.oracle.com/technetwork/articles/systems-hardware-architecture/oracle-db-smart-flash-cache-175588.pdf
- Sun Flash Accelerator F20
  - http://www.oracle.com/us/products/servers-storage/storage/disk-storage/043966.html
- Sun Storage F5100 Flash Array
  - http://www.oracle.com/us/products/servers-storage/storage/disk-storage/043967.html
- Fusion-io ioDrive
  - http://fusionio.biz/platforms/iodrive/

Credits

I'd like to thank the following people:

- Joel Goodman – Oracle University UK
  - Technical contributer and reviewer
- Christian Spaans - MCX Administration Services BV NL
  - Technical contributer
- Bernard van Aalst - MCX Administration Services BV NL
  - Technical contributer
- Hester Marijnissen
  - Editor

QUESTIONS & ANSWERS

And Finally

Thank you for your kind attention!

For a copy of my demonstration scripts email me at:

Harald.van.Breederode@oracle.com

Remember: prutser.wordpress.com