Database In Memory Machine Exadata X3-2 – How does it work?

Konrad HÄFELI
Senior Solution Manager
Infrastructure Engineering
Our company

Trivadis is a market leader in IT consulting, system integration, solution engineering and the provision of IT services focusing on ORACLE® and Microsoft technologies in Switzerland, Germany and Austria.

We offer our services in the following strategic business fields:

Trivadis Services takes over the interacting operation of your IT systems.
With over 600 specialists and IT experts in your region

12 Trivadis branches and more than 600 employees

200 Service Level Agreements

Over 4,000 training participants

Research and development budget: CHF 5.0 / EUR 4 million

Financially self-supporting and sustainably profitable

Experience from more than 1,900 projects per year at over 800 customers
- Hype?
- Buzzword?
- Megatrend?
- Commodity?
- Clearly defined?
Gartner Hype Cycle for Emerging Technologies, Source: Gartner.com August 2013
IN-MEMORY in Oracle context

- **Times-Ten**
  - *Oracle TimesTen In-Memory Database* stores data in application tier *main memory*, and with no network latency or disk I/O, transactions take just *microseconds* and complex analytic queries happen at the speed of thought.
  - *Oracle In-Memory Database Cache* enables database applications to *selectively cache* critical subsets of Oracle Database tables into Oracle TimesTen In-Memory Database to *improve* application response time.

- **Exalytics In-Memory Machine**
  - Combination of different database/analytics technologies

- **Exadata X3-2 Database In-Memory Machine (OOOW12)**
- **Database 12c In-Memory Option (OOOW13)**

---

*21st November 2013*

DOAG Jahreskonferenz - In Memory Database Machine Exadata X3-2 – How does it work?
Mass Memory Hierarchy for Extreme Performance

- Changes **from disk based with memory for acceleration to primarily memory based with disks** for capacity
  - X3 automatically moves all active user data to memory for extreme performance
- DRAM memory expanded to 2 or 4 TB for hottest data
- Flash memory **22 TB per rack**
- More than one Million SQL random I/Os per second for OLTP

**→ IN-MEMORY ???**
IN-MEMORY or not IN-MEMORY that’s the question…

- Kevin Closson, former Exadata Lead Architect…
- Discussion about what is IN-MEMORY and what not…

Oracle didn’t issue a press release about Exadata “In-Memory Database.” No, not “In-Memory Database” but “Database In-Memory” and the distinction is quite important…

In short, Oracle’s “Database In-Memory” data accesses are well beyond 40-fold slower than DRAM. But it’s not so much about device service times. It’s about overhead…
Definition of In-Memory Database (IMDB)

- As discussed in the blog of @flashdba
  - http://flashdba.com/category/blog/
- According Oracle Times Ten Documentation there are two statements

**IMDB Fundamental Requirement #1:**
In Memory Databases fit entirely in physical memory

**IMDB Fundamental Requirement #2:**
In Memory Databases are fast because they do not have complex code paths for dealing with data located on storage.
Oracle gave up ;-)
No matter how you name it...

There is an interesting Exadata Database Machine functionality:

Oracle Exadata Smart Flash Cache

- Transparently cache 'hot' read and write data to fast solid-state storage, improving query response times and throughput

Therefore I have to change the topic of this speech...
Oracle Exadata Flash Cache X3-2 – How does it work?

Konrad HÄFELI
Senior Solution Manager
Infrastructure Engineering
AGENDA

1. Introduction to Exadata Smart Flash cache
2. Flash cache
3. Flash logging
4. Flash Diskgroup
5. Test-Results with flash cache
6. Conclusion
Exadata X3-2 Storage Server
Introduction to Flash cache on Exadata

- Sun Flash Accelerator F40 PCIe Card
  - 4 per Storage Server
    - Each 400GB → 1.6TB
- 56 cards per full Rack → 22.4 TB of flash memory
- Oracle is using flash PCIe cards in Exadata – not flash disks
  - No limiting slow disk controller performance
  - Scale out across the PCI cards and Storage Cells
- Implemented directly in the Oracle Exadata Storage Server
  - holds frequently accessed data in very fast flash storage
  - while most of the data is kept in very cost effective disk storage
Three main functionalities

- Caches database objects in flash memory
  - Automated Management
  - User Management → Pinning Objects in the Flash Cache

- Exadata Smart Flash Logging
  - Flash for Database Logging

- Creating Flash Disks Out of the Flash Cache

- Btw. mission Critical Availability of the Exadata Smart Flash Cache
  - Automatical detection and offlineing
## Performance figures

<table>
<thead>
<tr>
<th></th>
<th>Exadata Database Machine X3-8 and X3-2 Full Rack</th>
<th>Exadata Database Machine X3-2 Half Rack</th>
<th>Exadata Database Machine X3-2 Quarter Rack</th>
<th>Exadata Database Machine X3-2 Eighth Rack</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exadata Smart Flash Cache</strong></td>
<td>22.4 TB</td>
<td>11.2 TB</td>
<td>4.8 TB</td>
<td>2.4 TB</td>
</tr>
<tr>
<td><strong>Disk Data Bandwidth</strong></td>
<td>Up to 25 GB/sec</td>
<td>Up to 12.5 GB/sec</td>
<td>Up to 5.4 GB/sec</td>
<td>Up to 2.7 GB/sec</td>
</tr>
<tr>
<td>- High Performance SAS</td>
<td>18 GB/sec</td>
<td>9.0 GB/sec</td>
<td>4.0 GB/sec</td>
<td>2.0 GB/sec</td>
</tr>
<tr>
<td>- High Capacity SAS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Flash Data Bandwidth</strong></td>
<td>Up to 100 GB/sec</td>
<td>Up to 50 GB/sec</td>
<td>Up to 21.5 GB/sec</td>
<td>Up to 10.7 GB/sec</td>
</tr>
<tr>
<td>- High Performance SAS</td>
<td>93 GB/sec</td>
<td>46.5 GB/sec</td>
<td>20 GB/sec</td>
<td>10 GB/sec</td>
</tr>
<tr>
<td>- High Capacity SAS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Database Flash Cache 8K Read IOPS</strong></td>
<td>Up to 1,500,000</td>
<td>Up to 750,000</td>
<td>Up to 375,000</td>
<td>Up to 187,000</td>
</tr>
<tr>
<td><strong>Database Flash Cache 8K Write IOPS</strong></td>
<td>Up to 1,000,000</td>
<td>Up to 500,000</td>
<td>Up to 250,000</td>
<td>Up to 125,000</td>
</tr>
<tr>
<td><strong>Database Disk IOPS</strong></td>
<td>Up to 50,000</td>
<td>Up to 25,000</td>
<td>Up to 10,800</td>
<td>Up to 5,400</td>
</tr>
<tr>
<td>- High Performance SAS</td>
<td>28,000</td>
<td>14,000</td>
<td>6,000</td>
<td>3,000</td>
</tr>
<tr>
<td>- High Capacity SAS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exadata Disks Storage Layout

Physical disks map to a Cell Disks

- Cell Disks partitioned into one or multiple Grid Disks
- ASM diskgroups created from Grid Disks
- Transparent above the ASM layer
Physical cards map to a Cell Disks

Cell Disks are either

- Added to FlashCache
- Partitioned as Griddisks to be visible in ASM for normal storage
ASM failure groups are used to protect against cell failures.
AGENDA

1. Introduction to Exadata Smart Flash cache
2. Flash cache
3. Flash logging
4. Flash Diskgroup
5. Test-Results with flash cache
6. Conclusion
Intelligent Caching

- Exadata Smart Flash Cache understands different types of database I/O
  - Frequently accessed data and index blocks are cached
  - Control file reads and writes are cached
  - File header reads and writes are cached
  - DBA can influence caching priorities

- I/Os to mirror copies are not cached
- Backup-related I/O is not cached
- Data Pump I/O is not cached
- Data file formatting is not cached
- Table scans do not monopolize the cache
(Default) read/write operation

- Object tagging `CELL_FLASH_CACHE`
  - DEFAULT/KEEP/NONE

```
alter table <table_name> storage (cell_flash_cache NONE);
```

- Database Cache hint
  - CACHE/NOCACHE/EVICT

→ decisions about which data is suitable for caching and which is not
## (Default) read/write operation (2)

<table>
<thead>
<tr>
<th>WRITE</th>
<th>READ (cached)</th>
<th>READ (uncached)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Send tagged data to disk</td>
<td>- CELLSRV checks cache via hash table</td>
<td>- CELLSRV checks cache via hash table</td>
</tr>
<tr>
<td>- Get acknowledge</td>
<td>- If in cache, cache lookup</td>
<td>- If NOT in cache, disk read</td>
</tr>
<tr>
<td>- If suitable, put data to cache</td>
<td></td>
<td>- If suitable, put data to cache</td>
</tr>
</tbody>
</table>
Exadata Smart Flash Cache Write-Back

- Caches **Write I/Os in flash** in addition to Read I/Os
- Accelerates write intensive workloads
  - Frequently updated tables and indexes
  - 20X more write IOPS than disk on X3
- Database writes go directly to flash cache
  - Block is kept in cache until it LRUs out
  - Could be months or years
  - While it is cached, reads or writes will be serviced from cache
  - Block will eventually age out of cache and will be written to disk
- Smart Caching applies
  - For example RMAN Backup and Data Pump reads and writes are not cached
Flash Cache Write-Back support

- Exadata Software 11.2.3.2.0

- But check MOS 888828.1 and dig in...
- Minimum version required if using Write-back Smart Flash Cache
  - Patch 14522699 - Exadata Storage Server software 11.2.3.2.1
  - Requires the Grid Infrastructure and Database home software to be 11.2.0.3.9

→ **Feature works on all hardware generations** (Exadata V2 and X2)

```
[root@dm01cel01 ~]# imageinfo

Kernel version: 2.6.32-400.11.1.el5uek #1 SMP Thu Nov 22 03:29:09 PST 2012 x86_64  
Cell version: OSS_11.2.3.2.1_LINUX.X64_130109  
Cell rpm version: cell-11.2.3.2.1_LINUX.X64_130109-1
```
Write-Back cache considerations

- Write Caching primarily benefits very high-write workloads
  - Applications that are not bottlenecked on writes will see little or no benefit from the extra writes enabled by write-back

- Unlike reads, improved write latency doesn’t help the application since writes are performed in background by DBWR

- Write IOPs are now so high that usually another bottleneck is reached before writes are maxed – application, CPU usage, mid-tier, latches, etc

- Monitor write hits
  - Read write on disk and flash celldisks at cell stats level
  - DB stat “Physical Write Requests Optimized” will be added in 11.2.0.4 → 28th of August 2013
Enabling Flash Cache Write-Back

CellCLI> list cell attributes flashcachemode
WriteThrough

CellCLI> drop flashcache
Flash cache ... successfully dropped

CellCLI> alter cell shutdown services cellsrv
Stopping CELLSRV services...
The SHUTDOWN of CELLSRV services was successful.

CellCLI> alter cell flashCacheMode = WriteBack
Cell ... successfully altered

CellCLI> alter cell startup services cellsrv
Starting CELLSRV services...
The STARTUP of CELLSRV services was successful.

CellCLI> create flashcache all
Flash cache ... successfully created
AGENDA

1. Introduction to Exadata Smart Flash cache
2. Flash cache
3. Flash logging
4. Flash Diskgroup
5. Test-Results with flash cache
6. Conclusion
Exadata Smart Flash Logging: Flash for Database Logging

THE Fact:

- OLTP workload, Logwriter Speed matters!
  - fast response time for database log writes is crucial
    → Low avg log file sync wait

THE Problem:

- Occasional “hiccups” in performance
  - On disks possible
  - On flash due to erase cycles or wear leveling

The Solution:

- Smart Flash Logging
  - Exadata Storage Software version 11.2.2.4
  - Oracle Database version 11.2.0.2 with Bundle Patch 11
Smart Flash Logging functionality

- Mirrored redo logfiles
  - Wait for the slower device (disk/disk, disk/flash)
- Default 512MB of flash allocated
  - Temporary storage
  - Negligible to the total amount of flash
- Redologs still in full size on Disk
- Faster acknowledge wins
- Fully transparent
Transparent functionality

- Best practices still the same
  - Number/size redologs

- No intervention necessary
  - Default on
  - Handles crash/recovery

- End user transparency

- Just one difference

→ Consistently low latency for redo log writes
Seeing is believing!

- Monitoring cell statistics type ‘FLASHLOG’

CellCLI> LIST METRICDEFINITION WHERE objectType = 'FLASHLOG'
  FL_DISK_FIRST
  FL_EFFICIENCY_PERCENTAGE
  FL_EFFICIENCY_PERCENTAGE_HOUR
  FL_FLASH_FIRST ...

CellCLI> LIST METRICHISTORY WHERE objectType = 'FLASHLOG' AND metricValue != 0 AND name like 'FL_EFFICIENCY_PERCENTAGE.*' -
  ATTRIBUTES name, metricObjectName, metricValue, collectionTime

<table>
<thead>
<tr>
<th>FL_EFFICIENCY_PERCENTAGE</th>
<th>FLASHLOG 100 %</th>
<th>2013-09-25T22:59</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL_EFFICIENCY_PERCENTAGE_HOUR</td>
<td>FLASHLOG 100 %</td>
<td>2013-09-25T22:59</td>
</tr>
<tr>
<td>FL_EFFICIENCY_PERCENTAGE</td>
<td>FLASHLOG 100 %</td>
<td>2013-09-25T23:00</td>
</tr>
<tr>
<td>FL_EFFICIENCY_PERCENTAGE_HOUR</td>
<td>FLASHLOG 100 %</td>
<td>2013-09-25T23:00</td>
</tr>
<tr>
<td>FL_EFFICIENCY_PERCENTAGE</td>
<td>FLASHLOG 100 %</td>
<td>2013-09-25T23:01</td>
</tr>
<tr>
<td>FL_EFFICIENCY_PERCENTAGE_HOUR</td>
<td>FLASHLOG 100 %</td>
<td>2013-09-25T23:01</td>
</tr>
</tbody>
</table>

21st November 2013
What if…

For consolidation platforms check and change the size

```
CellCLI> CREATE FLASHLOG ALL

CellCLI> CREATE FLASHLOG ALL SIZE=4g

CellCLI> CREATE FLASHLOG CELLDISK='fd1,fd2,fd3,fd4'
```

Exadata IORM (IO Resourcemanager)

→ enable or disable Smart Flash Logging for the different databases
AGENDA

1. Introduction to Exadata Smart Flash cache
2. Flash cache
3. Flash logging
4. Flash Diskgroup
5. Test-Results with flash cache
6. Conclusion
Flash disks

- Default all flash is assigned to Flash Cache (resp. Flash Logging)
- Optionally, a portion of the cache can be reserved and used as logical flash disks
  - treated like any Exadata cell disk in the Exadata cell
  - Max 4 cell disks per flash card (16 per cell)
  - Same sizing rules as on physical cell disks
Create flash griddisk (1)

don flashlog;
drop flashcache;

create griddisk FLS_CD_00_dm01ce101 celldisk=FD_00_dm01ce101
create griddisk FLS_CD_01_dm01ce101 celldisk=FD_01_dm01ce101
create griddisk FLS_CD_02_dm01ce101 celldisk=FD_02_dm01ce101
create griddisk FLS_CD_03_dm01ce101 celldisk=FD_03_dm01ce101
...

create flashlog celldisk='FD_04_dm01ce101,FD_05_dm01ce101, -
FD_06_dm01ce101,FD_07_dm01ce101'
create flashcache celldisk='FD_04_dm01ce101,FD_05_dm01ce101, -
FD_06_dm01ce101,FD_07_dm01ce101'
Create flash diskgroup (2)

```
SQL> create diskgroup FLASH_DG external redundancy disk
    'o/192.168.10.3/FLS_CD_00_dm01cel01,',
    'o/192.168.10.3/FLS_CD_01_dm01cel01,',
    'o/192.168.10.3/FLS_CD_02_dm01cel01,',
    'o/192.168.10.3/FLS_CD_03_dm01cel01','
    'o/192.168.10.4/FLS_CD_00_dm01cel02,',
    'o/192.168.10.4/FLS_CD_01_dm01cel02,',
    'o/192.168.10.4/FLS_CD_02_dm01cel02,',
    'o/192.168.10.4/FLS_CD_03_dm01cel02,',
    'o/192.168.10.5/FLS_CD_00_dm01cel03,',
    'o/192.168.10.5/FLS_CD_01_dm01cel03,',
    'o/192.168.10.5/FLS_CD_02_dm01cel03,',
    'o/192.168.10.5/FLS_CD_03_dm01cel03,',
attribute
    'cell.smart_scan_capable'='TRUE',
    'compatible.asm'='11.2.0.3.0',
    'compatible.rdbms'='11.2.0.3'
    'au_size'='4M';
```
**Move Tablespace to flash Diskgroup**

```sql
rman> sql 'alter tablespace exatest offline';
rman> copy datafile 6 to '+FLASH_DG';
rman> copy datafile 7 to '+FLASH_DG';
rman> switch datafile 6 to copy;
rman> switch datafile 7 to copy;
rman> recover tablespace exatest;
rman> sql 'alter tablespace exatest online';
```

**Considerations**

- Check the benefit for setting up diskgroups on flash
- Check option with object pinning in flash cache
AGENDA

1. Introduction to Exadata Smart Flash cache
2. Flash cache
3. Flash logging
4. Flash Diskgroup
5. Test-Results with flash cache
6. Conclusion
Test-Setup

- RAC Database on two nodes
- 1Gb SGA (to reduce caching on it and force more I/O on HD or FD)
- Connection over Service Name, both configured as preferred Instances for distribute the load
- Swingbench as Benchmarking Tool
- Stress Test Benchmarking Configuration (Adaption of 70/30 Ratio between Write/Read Operation)
- Always 4 Runs per Testing Szenario each Runs 5 Min
Test-Szenarios

- **WA:** no writeback, no flashcache, no flashlog, no cell keeping (Baseline)
- **FCO:** only flashcache
- **FLO:** only flashlog
- **FCLO:** flashcache and flashlog (exadata default configuration)
- **…WB:** including writeback
- **…OP:** Inc ding Object Pinning
- **…DG:** asm diskgroup with flashcache, flashcache and flashlog
Graphically result sets

**Avg Transactions per Second**

- WA
- FCO
- FCL
- FCO and FCL
- FCO, FCL, and WB
- FCO, FCL, WB, and OP
- FCO, FCL, WB, OB, and FC DG

**Total Completed Transactions**

- WA
- FCO
- FCL
- FCO and FCL
- FCO, FCL, and WB
- FCO, FCL, WB, and OP
- FCO, FCL, WB, OB, and FC DG

**Max. Transaction Rate**

- WA
- FCO
- FCL
- FCO and FCL
- FCO, FCL, and WB
- FCO, FCL, WB, and OP
- FCO, FCL, WB, OB, and FC DG

**CPU Usage**

- Avg Sys CPU
- Avg User CPU
Result-Analysis

1. Flash Logging increases the performance by 24%
2. Default usage of flash cache and log increase by 34%
3. Enabling Write-Back Cache increases another 13% to the default
4. Object pinning had no measurable benefit over normal caching
   - But significant lower CPU load
5. Flash diskgroups can be negative because of the missing resources for the cache
AGENDA

1. Introduction to Exadata Smart Flash cache
2. Flash cache
3. Flash logging
4. Flash Diskgroup
5. Test-Results with flash cache
6. Conclusion
Conclusion

- Exadata X3-2 is NOT an IN-Memory Machine ;-)  
- But an impressive flash cache based Database platform  
- There is a lot of intelligent functionality inside  
  - Flash caching  
  - Flash logging  
  - Write-back caching  
  - Flash cache diskgrouping  
- All of them customizable/tunable  
- As most things on Exadata, the defaults are good!  
- Write-back cache functionality can be considered to switch on
... but if you really wanna have an IN-MEMORY Exadata:
Questions and answers ...

Konrad Häfeli
Senior Solution Manager
Infrastructure Engineering

Konrad.Haefeli@trivadis.com
Trivadis an der DOAG

Ebene 3 - gleich neben der Rolltreppe

Wir freuen uns auf Ihren Besuch.

Denn mit Trivadis gewinnen Sie immer.