

Improving Performance With 12c In-Memory Option

Prepared by: Fong Zhuang & Sergiy Smyrnov

Session ID: CON4349

©2017 Walgreen Co. All rights reserved.

Walgreens
at the corner of **happy** & **healthy**[®]

Member of Walgreens Boots Alliance

Speaker

Fong Zhuang

- ❖ Database Architect and Team Lead
- ❖ 19+ years with Walgreens
- ❖ My Specialty & Interest
 - ❖ Oracle Performance Tuning and High Availability
 - ❖ Oracle E-Business Suit (EBS) and Oracle Business Intelligence Enterprise Edition (OBIEE)
 - ❖ Oracle Engineered System
 - ❖ Data Warehouse Implementation
- ❖ Oracle Certified Professional (OCP)



Speaker

Sergiy Smyrnov

- ❖ Lead Database Architect
- ❖ 20+ years of experience with Database Technologies
- ❖ My Areas of expertise
 - ❖ Oracle Engineered Systems
 - ❖ High Availability/DR implementations for database technologies
 - ❖ Data Warehouse/DSS and OLTP/E-commerce database systems
 - ❖ Database automation

<https://www.linkedin.com/in/sergiy-smyrnov-a2248728/>



Walgreens

- ❖ Included in the Retail Pharmacy USA Division of Walgreens Boots Alliance, Inc., the first Global Pharmacy-Led Health and Wellbeing Enterprise
- ❖ One of the nation's largest drugstore chains
- ❖ More than 8,175 stores
- ❖ Approximately 400 Walgreens stores offer Healthcare Clinic or other provider retail clinic services
- ❖ Focused on enhancing our customers' beauty shopping experience



Agenda

- Row-major Oriented vs Column Oriented DBMS
- Introduction of Oracle Dual Format SGA Architecture
- Case Studies in Walgreens
 - Case 1 OLTP EBS environments (Exadata)
 - Case 2 OLAP OBIEE Data Warehouse (ODA vs. Exadata)
- Release 12.2 In-Memory enhancement
- Walgreens roadmap to use In-Memory features



Row-major Oriented vs Column Oriented DBMS

Row-major
Oriented
DBMS



Column
Oriented
DBMS



CHALLENGE: Accelerate the performance of both OLAP and OLTP in the same database.

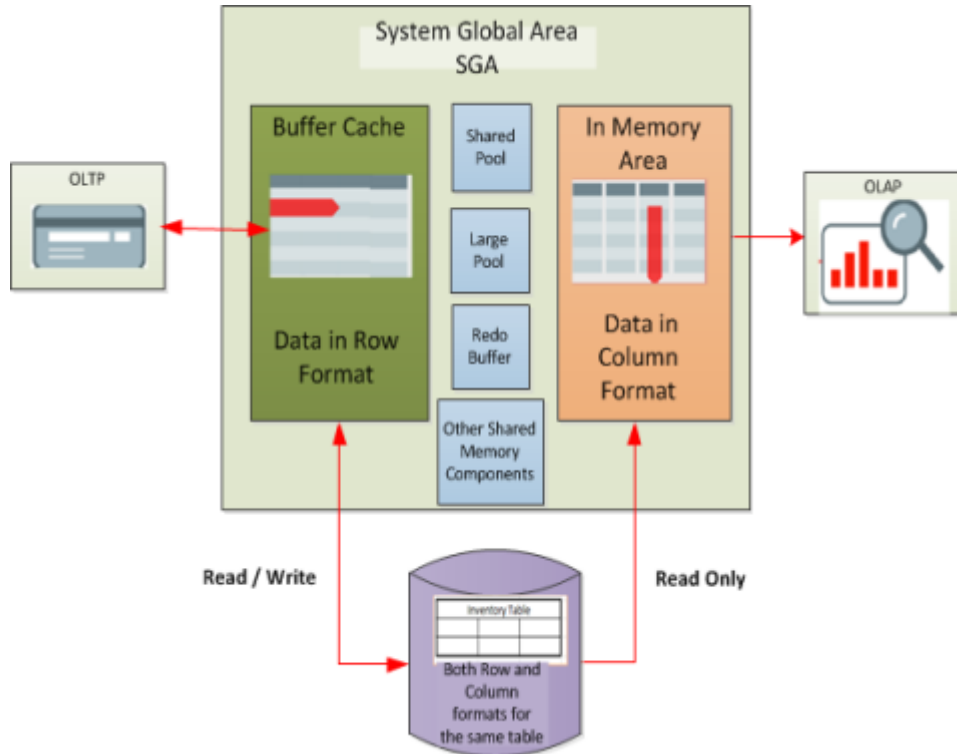
Traditional Row-major DBMS → OLTP workloads.

Columnar Oriented DBMS → A proven practice for OLAP / Data Warehouse

Oracle innovative OLTP solution

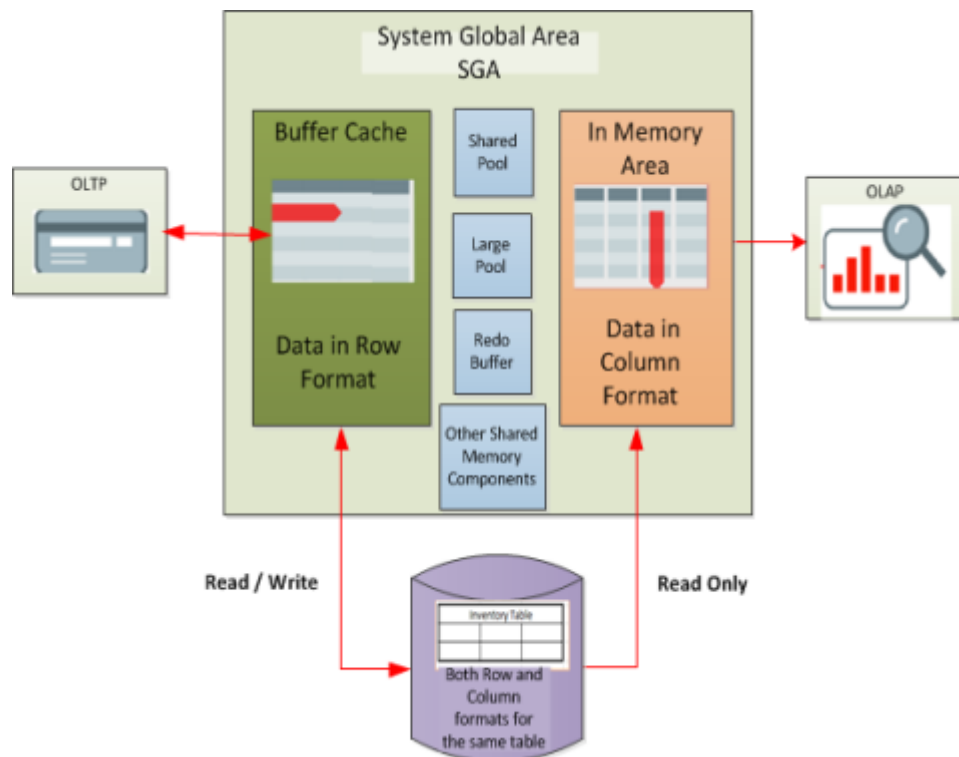


Innovative OLTP Solution - Dual Format Architecture



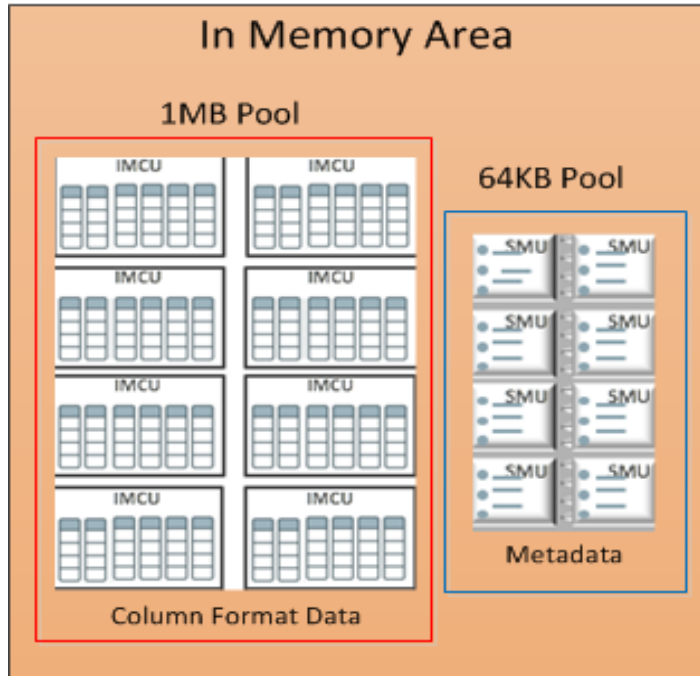
- **New In-Memory Area option in Oracle Database 12c Release 1 (12.1.0.2)**
- **Static SGA pool in special columnar format for OLAP.**
- **Speeding up analytical queries**
- **Buffer Cache in row format for OLTP**
- **Same data stored in both Buffer Cache & In-Memory Area in different formats.**

Transactionally Consistent



- ❑ Data stored in disk in Row format
- ❑ DML <---> Buffer Cache row store.
- ❑ Read-only → In-Memory Column Store.
- ❑ Transaction (inserts, updates, and deletes) committed → Both stores.
- ❑ The population includes a transformation from row to columnar format and it consumes CPU.

In-Memory Area Structure



- **1MB-Pool IMCU (In-Memory Compression Unit)**
 - Logical unit of storage
 - Roughly equivalent to an extent within a tablespace.
- **64KB-Pool SMU (Snapshot Metadata Unit)**
 - Metadata about the IMCU & transactional info
 - Records min and max values
 - Serves as an In-Memory Storage Index
- Each IMCU maps to a SMU.
- **NO Least-Recently-Used-Mechanism (LRU)**

Case Study in Walgreens Financial System

❑ Case 1 in EBS

- OLTP mix-workload
- Relational entity model in 3NF.

❑ Case 2 in OBIEE Data Warehouse

- OLAP
- Entity model star schema

❑ Lower environments in ODAs

❑ Performance test and PROD environments in Exadata



Case Study Details

Please refer to following paper for more details:

❖ **Published in IOUG Select:**

“Improving Performance With 12c In-Memory Option”

<http://select.ioug.org/blog/improving-performance-with-12c-in-memory-option> by Fong Zhuang



Case 1 OLTP - Walgreens Oracle EBS (Exadata)

Database Evaluation:

- POC Performance Test Environment
- Database host: Exadata X-3 Quarter Rack
- RAC (2 nodes)
- Host memory: 256GB / node
- 16 cores / node
- DB version 12.1.0.2



Leverage In-Memory Advisor

Use Active Session History (ASH), Automatic Workload Repository (AWR) and optionally SQL Tuning Sets (STS)

Determine which tables, partitions and sub-partitions to place In Memory.

Licensed as part of the Database Tuning pack.

Minimum DB version to run In-Memory Advisor is 11.2.0.3.

Implemented recommendations in Database 12.1.0.2 and above.



Leverage In-Memory Advisor(Cont.)

Where to get it and how to install it?

- MOS note: [1965343.1](#)
- Download imadvisor.zip from Oracle, copy to DB server-->unzip it
- Install in SQLPLUS with sysdba privilege
 - SQL> @instimadv.sql
- Run the IM advisor recommendation script with appropriate period
 - SQL> @imadvisor_recommendations.sql
- Review two outputs generated from the script



Case 1 OLTP - Walgreens Oracle EBS

In-Memory Advisor report highlights:

- A report based on 1 week of workload in EBS
- 67% DB time is doing analytics processing
- With 535.2GB In-Memory Size, the Estimated Analytics Processing Time Reduction is 79 hours.
- With 20GB of In-Memory, the potential time reduction is 14 hours.



Size **SGA** and **PGA** properly

Check MOS Note [1903683.1](#)

✓ Existing Single Instance:

$SGA_TARGET = \text{Existing } SGA_TARGET + INMEMORY_SIZE$

✓ Existing RAC:

$SGA_TARGET = \text{Existing } SGA_TARGET + INMEMORY_SIZE \times 1.1$

✓ If you are using parallel execution to reduce the possibility of spilling to TEMP on large joins and aggregations

$PGA_AGGREGATE_TARGET = PARALLEL_MAX_SERVERS * 2G$

Note:

- ✓ Additional space is required for the shared_pool as IMCS allocates additional locks from the shared pool
- ✓ Avoid “ORA – 4031” - unable to allocate string bytes of shared memory

Enable In-Memory Column Store

- ❑ Controlled by the parameter `INMEMORY_SIZE`
- ❑ Minimum size of 100MB
- ❑ In RAC DB, add additional 10% overhead to `INMEMORY_SIZE`.

```
SQL> alter system set sga_max_size = 70G scope=spfile sid='*';
```

```
SQL> alter system set sga_target = 70G scope=spfile sid='*';
```

```
SQL> alter system set inmemory_size = 30G scope=spfile sid='*';
```

- ❑ Restart the database to take effect.

```
SQL> show sga
```

```
Total System Global Area 7.5162E+10 bytes
Fixed Size                7652376 bytes
Variable Size             1.7985E+10 bytes
Database Buffers         2.4696E+10 bytes
Redo Buffers
In-Memory Area          3.2212E+10 bytes
```

```
SQL> show parameter inmemory_size
```

NAME	TYPE	VALUE
inmemory_size	big integer	30G

In-Memory Area Structure

- ❖ The pool information in v\$inmemory_area

```
SQL> select * from v$inmemory_area;
```

POOL	ALLOC_BYTES	USED_BYTES	POPULATE_STATUS	CON_ID
1MB POOL	2.5744E+10	0	DONE	0
64KB POOL	6422724608	0	DONE	0

- ❖ By default, 20% of the In-memory size is allocated to the 64KB pool.
- ❖ Most of the chances, 1MB pool exhausted but 64KB pool plenty of space.
- ❖ No mechanism automatically resize 1MB pool and 64KB pool.
- ❖ Option of modifying "**_inmemory_64k_percent**" parameter to reduce the size of the 64K pool **with Oracle support's approval.**

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Target the bottleneck :

- App team's request: One EBS batch job was taking 14 hours & 39 min. Any solutions?

Name	WAG Sales Tax Outbound Interface		
Operating Unit			
Parameters	1040, 2021, May-15		
Phase	Completed	Status	Normal
Date Submitted	05-JAN-2017 14:12:11	Requestor	2297722
Date Started	05-JAN-2017 14:12:30	Date Completed	06-JAN-2017 05:01:43
Completion Text	Normal completion		
Language	American English	Languages...	

14 hours 39 min

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

In-Memory advisor html report analysis:

Workload Database Usage

Total Database Time (Seconds)	Analytics Processing Time (Seconds)	Analytics Processing Percentage
59140	51520	87%

In-Memory Size	Percentage of Maximum SGA Size (50,00GB)	Estimated Analytics Processing Time Reduction (Seconds)	Estimated Analytics Processing Performance Improvement Factor
30.00GB	60%	17290	1.5X
1.852GB	4%	17290	1.5X
1.739GB	4%	17261	1.5X
1.667GB	3%	17261	1.5X
1.574GB	3%	17261	1.5X
1.481GB	3%	17261	1.5X
1.389GB	3%	17161	1.5X
1.296GB	3%	17161	1.5X
1.204GB	2%	17161	1.5X
1.111GB	2%	17161	1.5X
1.018GB	2%	17161	1.5X
948.1MB	2%	17161	1.5X
853.3MB	2%	17161	1.5X
758.5MB	1%	17161	1.5X
663.7MB	1%	17161	1.5X
568.8MB	1%	17161	1.5X
474.0MB	1%	17161	1.5X
379.2MB	1%	17161	1.5X
284.4MB	1%	17161	1.5X
189.6MB	0%	17155	1.5X
94.81MB	0%	17046	1.5X

If you add the objects to IMCS with 94.8MB in size, the time reduction is 4.7 hours.

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

In-Memory advisor html report analysis (Cont.)

Only 850 sec out of 51,520 sec (14.3 hours) of Total Analytics Processing Time

Per AWR, this SQL took avg 0.04 sec to run. It was executed 27,072 times.

*With 30.00GB, The **One** SQL Statement With Analytics Processing Benefit*

SQL Id	SQL Text	Analytics Processing Time Used (Seconds)	Estimated Analytics Processing Time Reduction (Seconds) With Unlimited Memory	Estimated Analytics Processing Performance Improvement Factor With Unlimited Memory	Estimated Analytics Processing Time Reduction (Seconds) With 30.00GB	Estimated Analytics Processing Performance Improvement Factor With 30.00GB
1223h53th9m3	SELECT XGL.LOCATION NUMBER STORE_NO ,XXGL.SALES TAX OUT OUND_PKG.GET_STATE_CODE_F (XGL.STATE) STAT...	850	7	1.0X	7	1.0X

Only 7 sec of reduction, Seriously?

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Populate Segments

→ 10 tables are recommended to populate into IMCS.

- Table Population Examples in Engineered System:

```
SQL> ALTER TABLE "GL"."GL_CODE_COMBINATIONS" INMEMORY  
duplicate all priority critical MEMCOMPRESS FOR QUERY LOW;
```

- Eligible Segments: Table, Partition, Subpartitions, Materialized Views, and tablespace.
- Tune parameter *inmemory_max_populate_servers* for segment population performance, but watch over the CPU usage carefully.
 - The default value is the less value: $1/2 \times \text{CPU thread count}$ **OR**
 $\text{PGA_AGGREGATE_TARGET} / 512\text{M}$

The IM column store compression Levels

```
SQL> ALTER TABLE "GL"."GL_CODE_COMBINATIONS" INMEMORY  
duplicate all priority critical MEMCOMPRESS FOR QUERY LOW;
```

- No Memcompress
- Memcompress For DML
- Memcompress For Query Low (default)
- Memcompress For Query High
- Memcompress For Capacity Low
- Memcompress For Capacity High



In-Memory Population Priority

```
SQL> ALTER TABLE "GL"."GL_CODE_COMBINATIONS"  
INMEMORY duplicate all priority critical  
MEMCOMPRESS FOR QUERY LOW;
```

- CRITICAL
- HIGH
- MEDIUM
- LOW
- NONE (Default)

When “In Memory” meets RAC

```
SQL> ALTER TABLE "GL"."GL_CODE_COMBINATIONS" INMEMORY  
duplicate all priority critical MEMCOMPRESS FOR QUERY LOW;
```

- For Non Engineered System, use “distribute” subclause
 - Distribute by rowid range
 - Distribute by partition
 - Distribute by subpartition
- Enable Auto DOP
 - Parallel_degree_policy=auto
- In Engineered System, option of using subclause “duplicate” or “duplicate all” to duplicate IMCUs into other nodes (mirroring).



Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Check the execution results after adding tables to In-Memory:

- The job completed in **1** hour and **9** minutes.
- Compare to **14** hours and **39** minutes of the previous timing, **12X performance improvement**.

Name	WAG Sales Tax Outbound Interface		
Operating Unit			
Parameters	1040, 2021, May-15		
Phase	Completed	Status	Normal
Date Submitted	20-JAN-2017 18:40:58	Requestor	2297722
Date Started	20-JAN-2017 18:41:24	Date Completed	20-JAN-2017 19:50:32
Completion Text	Normal completion		
Language	American English	Languages...	

1 hour 9 min

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Check the application execution results after adding tables to In-Memory (Cont.)

- Where was the performance improvement coming from ?

Reality: Per base AWR, after adding tables to In-Memory, it took 0.012 sec each time

Reality: It was executed 27,072 times

With 30.00GB, The One SQL Statement With Analytics Processing Benefit

SQL Id	SQL Text	Analytics Processing Time Used (Seconds)	Estimated Analytics Processing Time Reduction (Seconds) With Unlimited Memory	Estimated Analytics Processing Performance Improvement Factor With Unlimited Memory	Estimated Analytics Processing Time Reduction (Seconds) With 30.00GB	Estimated Analytics Processing Performance Improvement Factor With 30.00GB
1223h53fh9ht9	SELECT XGL.LOCATION NUMBER STORE_NO ,XXGL.SALES TAX_OUTB OUND_PKG.GET_STATE_CODE_F (XGL.STATE) STAT...	850	7	1.0X	7	1.0X

Reality: The total Processing Time Reduction is 758 Sec (12.4 Min)

Reality: The performance improvement factor is 3.3X

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Identify the key time saving SQL from AWR Report

SQL ordered by Elapsed Time

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 102.4% of Total DB Time (s): 54,650
- Captured PL/SQL account for 104.1% of Total DB Time (s): 54,650

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	% Total	% CPU	% IO	SQL Id	SQL Module	SQL Text
53,129.99	0		97.22	99.02	0.21	<u>44513hm880cwn</u>	XXGLSALESTAXOUT	BEGIN XXGL_SALES_TAX_OUTBOUND_...
50,582.91	538	94.02	92.56	99.23	0.00	<u>d5v6120ykm39r</u>	XXGLSALESTAXOUT	SELECT DISTINCT FFM.FLEX_VALUE...
1,223.84	1	1,223.84	2.24	98.47	0.57	<u>05s9358mm6vrr</u>		begin dbms_feature_usage_inter...
1,221.63	1	1,221.63	2.24	98.54	0.55	<u>duxwqgg8un28r</u>		BEGIN DBMS_FEATURE_AWR(:featur...
1,221.60	1	1,221.60	2.24	98.54	0.55	<u>6tw4m9dggqg2r</u>		SELECT /* DS_SVC */ /*+ dynami...
1,057.61	27,072	0.04	1.94	97.56	1.58	<u>1223h53fh9ht9</u>	XXGLSALESTAXOUT	SELECT XGL.LOCATION_NUMBER STO...
397.77	349	1.14	0.73	94.28	3.32	<u>dffkcnqfystw</u>	MMON_SLAVE	WITH MONITOR_DATA AS (SELECT I...
365.76	349	1.05	0.67	93.67	3.72	<u>fhf8upax5cxsz</u>	MMON_SLAVE	BEGIN sys.dbms_auto_report_int...
355.77	349	1.02	0.65	93.79	3.74	<u>0w26sk6t6gg98</u>	MMON_SLAVE	SELECT XMLTYPE (DBMS_REPORT.GET...
339.89	1	339.89	0.62	99.17	0.07	<u>73629tmfbjzrh</u>	DBMS_SCHEDULER	DECLARE job BINARY_INTEGER := ...

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

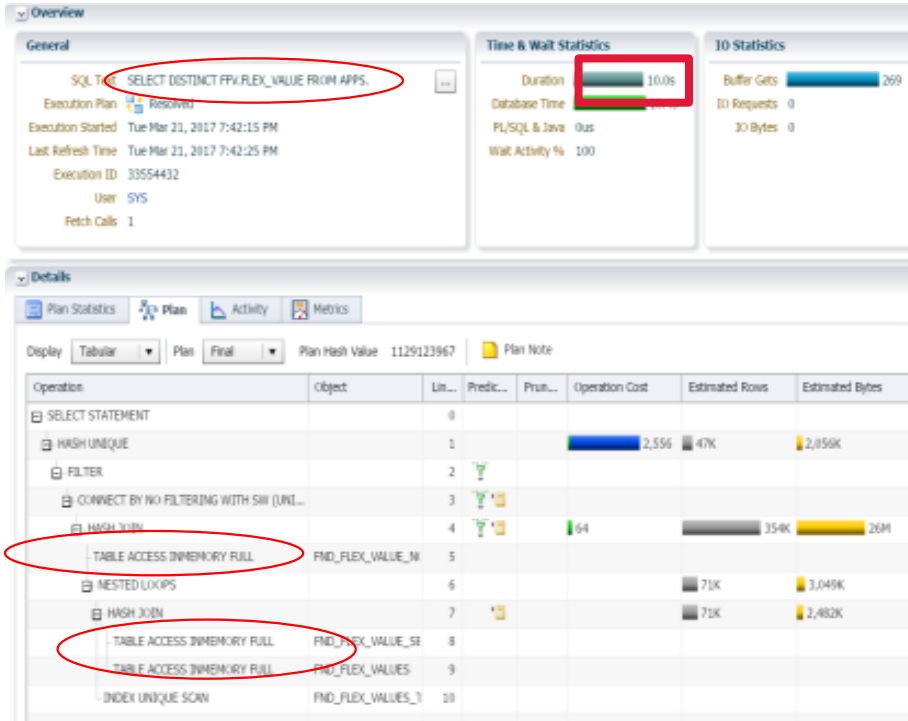
Top SQL performance evaluation in Wag Sales Tax Outbound job

SQL_ID	SQL Text	Executions	Average Elapsed Time per Exec (s)	Average Elapsed Time per Exec w tables In-Memory (s)	Time Reduced/Execution (s)	Performance Improvement Factor (Old Timing / New Timing)	Total time reduction(s)
1223h53f h9ht9	SELECT XGL.LOCATIO N_NUMBER STO...	27,072	0.04	0.012	0.028	3.33X	758 (s) OR 12.6 (m)
d5v6120y km39r	SELECT DISTINCT FFV.FLEX_VAL UE FROM APPS.FND_FL EX_VALUE_CH ILDREN_V FFV	538	94.02	2.16	91.86	47X	49496 (s) OR 825 (m) OR 13.7 (h)

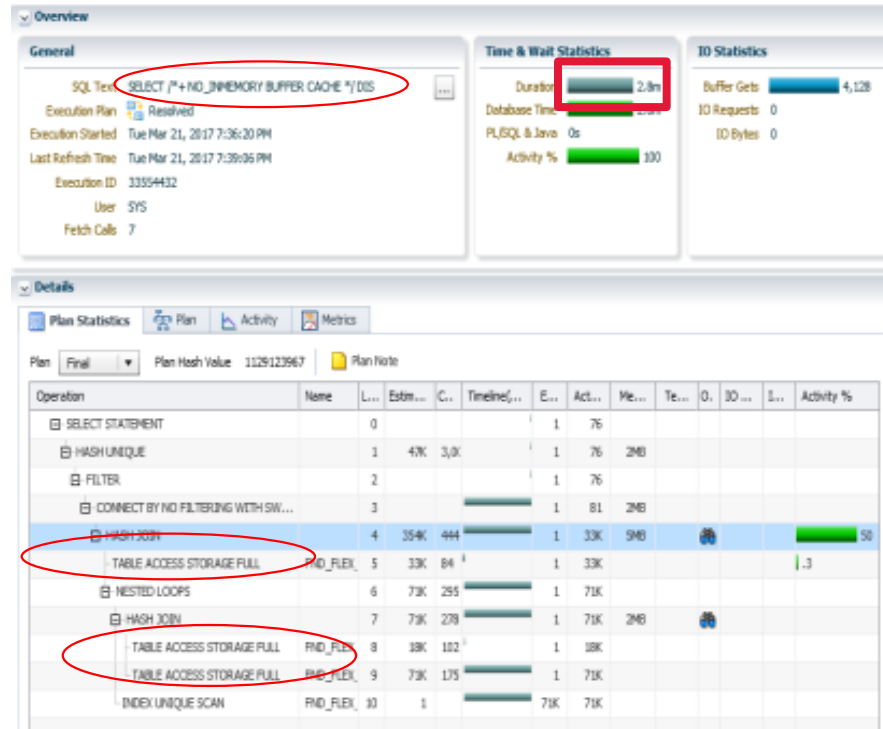
Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Top SQL2 Execution Plan Comparison from SQL Monitor Output

Column Store: In-Memory



Row Store: Buffer Cache



Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Test results with good performance improvement on major EBS jobs:

EBS Job Name	Table Disk Size	In-Memory Size	Processing Time w/o In-Memory	Processing Time with In-Memory	Time Reduced	Performance Improvement Factor (Old Timing / New Timing)
Outbound Interface Program	11GB	7GB	9 hours	5 hours	4 hours	1.8 X
Sales Tax Outbound Interface	8.5GB	1.8GB	14 hours and 39 minutes	1 hour 10 minutes	13 hours	12 X



Note: There were 36 indexes on 10 tables. The total index disk size is 2.3 GB. Analytic indexes can be dropped after adding tables into IMCS.

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Sample test results without performance improvement:

EBS Job Name	Disk Size	In-Memory Size	Processing Time w/o In-Memory	Processing Time with In-Memory	Time Reduced	Performance Improvement Factor (Old Timing / New Timing)
Warehouse Billing Journal Inbound Interface	89GB	15GB +	1 hour and 20 minutes	1 hour 28 minutes	N/A	N/A

Reasons with no improvement:

- One major table with 89GB which couldn't be fully populated to In-Memory.
- The In-Memory compression ratio is relatively high.

Case 1 OLTP - Walgreens Oracle EBS (Cont.)

Case 1 Summary:

- Observed time reduction from 1.8X to 12X on various jobs.
- In-Memory Advisor is efficient in identifying the candidate segments.
- The real performance could be much better than the In-Memory Advisor estimation.
- Analytic indexes can be dropped for potential space saving and DML contention reduction.
- No performance change on segments with high compression ratio.

Case 2 OLAP Oracle OBI Data Warehouse (ODA)

Database Evaluation:

- Proof of Concept Environment
- Database host:
Oracle Database Appliance (ODA X3-2)
- RAC (2 nodes)
- Host memory: 256GB/node
- 16 cores / node
- ODA is shared by multiple DBs
- DB version 12.1.0.2
- Plan to improve with INMEMORY_SIZE 50GB



Case 2 OLAP Oracle OBI Data Warehouse (ODA)

Sample test results without performance improvement:

Test Description	Disk Size	In-Memory Size	Tables NOT In Memory (HH:MM:SS.00)	Tables In Memory (HH:MM:SS.00)	Performance Improvement Factor (Old Timing / New Timing)
Count Fact Table W_GL_BALANCE_F	58GB	16GB	00:01:23.42	00:00:00.61	137 X
A Report Accesses Fact Table W_GL_BALANCE_F and 4 Small Dimension Tables with Many aggregates	58.6GB	16GB	00:02:48.88	00:00:09.12	18.5 X
Count Fact Table W_GL_OTHER_F	128GB	17GB	00:04:42.39	00:00:32.55	8.7 X

Case 2 OLAP Oracle OBI Data Warehouse (ODA)

Sample test results without performance improvement:

Test Description	Disk Size	In-Memory Size	Tables NOT In Memory (HH:MM:SS.00)	Tables In Memory (HH:MM:SS.00)	Performance Improvement Factor (Old Timing / New Timing)
A Report Accesses Fact Table W_GL_OTHER_F and 7 Small Dimension Tables	128.6GB	17GB	00:01:33.04	00:01:29.80	No Significant Change

Reasons with no improvement:

- One major table has 128.6 GB and the In-Memory Size is 17GB.
- The In-Memory compression ratio is relatively high.

Case 2 OLAP Oracle OBI Data Warehouse (Exadata)

Database Evaluation:

- Performance Test Environment
- Database host: Exadata X-3 Quarter Rack
- RAC (2 nodes)
- Host memory: 256GB/node
- 16 cores / node
- DB version 12.1.0.2
- Plan to improve with INMEMORY_SIZE 30GB



Case 2 Same SQL Performance Comparison Exadata vs ODA

Exadata X3-2

Test Description	Disk Size	In-Memory Size	Tables NOT In Memory (HH:MM:SS.00)	Tables In Memory (HH:MM:SS.00)	Performance Improvement Factor
Count Table W_GL_BALANCE_F	50.62GB	13.65GB	00:00:4.57	00:00:01.08	4.23 X
A Report Accesses Fact Table W_GL_BALANCE_F	51.2GB	14GB	00:00:22.47	00:00:07.34	3.06 X
Count Table W_GL_OTHER_F	128GB	13.56GB	00:00:04.22	00:00:01.03	4.09 X
A Report Accesses Fact Table W_GL_OTHER_F	128.6GB	13.60GB	00:00:05.19	00:00:06.80	No Significant Change

ODA X3-2

Test Description	Disk Size	In-Memory Size	Tables NOT In Memory (HH:MM:SS.00)	Tables In Memory (HH:MM:SS.00)	Performance Improvement Factor
Count Table W_GL_BALANCE_F	58GB	16GB	00:01:23.42	00:00:00.61	137 X
A Report Accesses Fact Table W_GL_BALANCE_F	58.6GB	16GB	00:02:48.88	00:00:09.12	18.5 X
Count Table W_GL_OTHER_F	128GB	17GB	00:04:42.39	00:00:32.55	8.7 X
A Report Accesses Fact Table W_GL_OTHER_F	128.6GB	17GB	00:01:33.04	00:01:29.80	No Significant Change

Case 2 OLAP Oracle OBI Data Warehouse

Case 2 Summary:

- Promising performance improvement in both ODA and Exadata
- Exadata provides large Flash Cache, Smart Scan, and HCC.
- Recommend to keep the very hot data in In-Memory column store, keep the warm data to Exadata Flash Cache, and compress the cold data with Exadata HCC.
- For either environment, we expect to add more memory for further performance improvement.

Exciting Future



Is that all? Can we do more with In-Memory?

Yes , we can (in 12cR2)!!!

- ILM (Information Lifecycle Management) for automated data policy management for IMCS in Exadata + 12cR2
- IMCS FastStart Area in 12cR2
- IMCS in Oracle Active Data Guard in 12cR2
- Improvements of IMCS for RAC services in 12cR2
- In-Memory Expressions in 12cR2

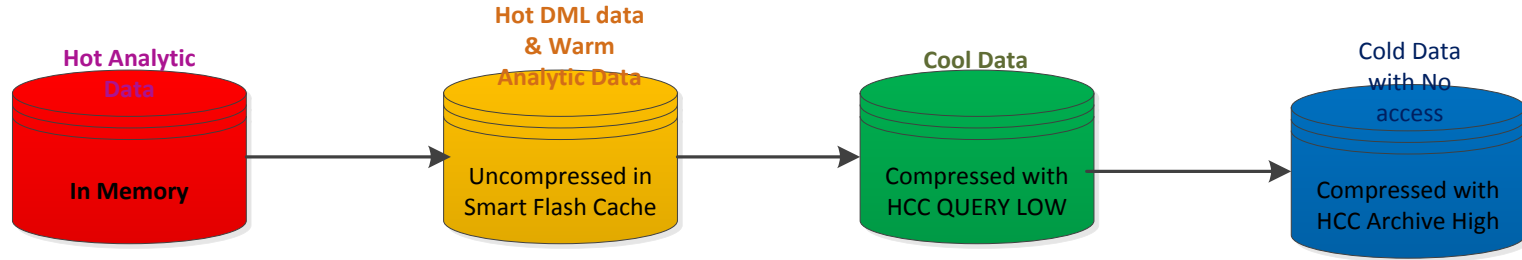


ILM for ADO management for IMCS in Exadata + 12cR2

The Oracle Information Lifecycle Management (ILM) has the capability of distinguish between “hot” (frequently accessed) and “cold” (infrequently accessed) data and set policy on tables and segments.

In 12.2, DBAs can set policies to mark an object as a candidate to be populated into In-Memory column store (INMEMORY) and in Exadata additionally in Flash-Cache (CELLMEMORY) to being able to use ADO policies for WARM/COLD data.

This significantly enhance and automate further the lifecycle of your core data within a database.



Note that to in-memory ADO is only supported at the segment level



ILM ADO INMEMORY options

Use *ILM ADD POLICY* syntax with

- SET INMEMORY
- MODIFY INMEMORY
- NO INMEMORY

Alter MEMCOMPRESS for INMEMORY level

- After <time> of no access
- After <time> of creation
- After <time> of no modification
- On <user defined Boolean function>

Examples:

```
ALTER TABLE rx_fill ILM ADD POLICY NO INMEMORY after 3 month of creation;  
ALTER TABLE rx_fill ILM MODIFY POLICY NO INMEMORY after 7 days of no access;  
ALTER TABLE rx_fill ILM ADD POLICY MEMOCOMPRESS FOR QUERY LOW AFTER 30 days of creation;  
ALTER TABLE rx_fill ILM ADD POLICY MEMOCOMPRESS FOR QUERY HIGH AFTER 45 days of no  
modification;
```



Exadata CELLMEMORY option

CELLMEMORY works similar to a 12.1.0.2 columnar cache but when used with INMEMORY on Exadata is enabled by default to extend columnar cache format.

You can overwrite (enable,disable, change compression level) default and use CELLMEMORY feature on tables, partitions, sub partitions, and materialized views.

The MEMCOMPRESS clause supports FOR [QUERY|CAPACITY] [LOW| HIGH].

```
ALTER TABLE member CELLMEMORY;  
ALTER TABLE member NO CELLMEMORY;  
ALTER TABLE sales PARTITION Q1_2017 CELLMEMORY MEMCOMPRESS FOR QUERY LOW;
```

Note that this feature requires the 12.2 release of Exadata Storage Server software, but works with the 12.1 release of the database. Use of the 12.2 release of the database enables additional performance optimizations for offloaded CELLMEMORY scan processing.



In-Memory FastStart Area in 12cR2

A FastStart area is a designated tablespace where IM FastStart stores and manages data for INMEMORY objects.

Tablespace should be 2x size of INMEMORY_SIZE area

Oracle Database manages the FastStart tablespace without DBA intervention.

It can relieve the CPU overhead of population at the cost of additional disk space.

- In-memory Fast Start syntax:

```
exec DBMS_INMEMORY_ADMIN.FASTSTART_ENABLE('<fstart_tbs_name>');
```

- To check the size/status of Fast Start tablespace usage use view

V\$INMEMORY_FASTSTART_AREA

Note: the actual FastStart IM data is written in LOB segment in that tablespace.



In-Memory in Oracle Active Data Guard in 12cR2

In 12cR2 you now have option to enable IM column store on the standby Active Dataguard database (commonly a lot of applications usually take advantage of Active Dataguard for reporting purposes), or on both the primary and standby databases.

There are three configuration options to consider:

- Identical column stores in both primary and standby databases
- Column store in just the standby database
- Different column store configurations between the primary and standby databases

Those configurations are managed by the combination of 3 settings:

- `INMEMORY_SIZE` (primary, standby)
- `INMEMORY_ADG_ENABLED` is set to true on the standby instance (if needed)
- The `INMEMORY` with additional `DISTRIBUTE FOR SERVICE` clauses for `IN-MEMORY` objects designated for the standby role services



Improvements of In-Memory for RAC services in 12cR2

In Oracle 12c R2 it become much easier to selectively populate objects to specific IM column stores in a RAC environment by taking advantage of the new DISTRIBUTE FOR SERVICE subclause when defining INMEMORY option.

This is a secondary use of that sub-clause after INMEMORY on Active Dataguard.

- Example:

```
srvctl add service -db db1 -service rx_srv -preferred "inst1"  
srvctl add service -db db1 -service catalog_srv -r preferred "inst2"
```

```
ALTER TABLE rx_fill INMEMORY PRIORITY MEDIUM DISTRIBUTE FOR SERVICE rx_srv;
```

```
ALTER TABLE drugs INMEMORY PRIORITY HIGH DISTRIBUTE FOR SERVICE catalog_srv;
```



In-Memory Join Groups in 12cR2

With Oracle Database 12.2., it is possible to declare that two tables will be joined on a set of columns using a new DDL construct known as a *Join Group*.

This enables join processing optimizations based on updated common dictionary so that joins can be performed on compact dictionary codes instead of on values.

Note that IM segments need to be re-populated after creation of Join Groups.

- Examples:

```
CREATE INMEMORY JOIN GROUP rx_fill_jg1(rx_fill(patient_id),member(patient_id));  
  
CREATE INMEMORY JOIN GROUP rx_fill_jg2(rx_fill(drug_id),drug(drug_id));  
  
CREATE INMEMORY JOIN GROUP rx_fill_jg3(rx_fill(rx_id),invoice(rx_id));
```

You can query `USER_JOINGROUPS` view to see Join Group definitions.

In order to determine if Join Group(s) are actually in use you need to check a SQL Monitor Active Report and click on the “eye glasses” icon for hash joins and check that “Columnar Encodings Leveraged” is showing value “1”.



In-Memory Expressions in 12cR2

In-Memory Expressions give ability to materialize simple expressions and store them in the In-Memory column store so that they only have to be calculated once.

They are treated like any other column in the IM column store so the database can scan and filter those columns and take advantage of all Database In-Memory query optimizations

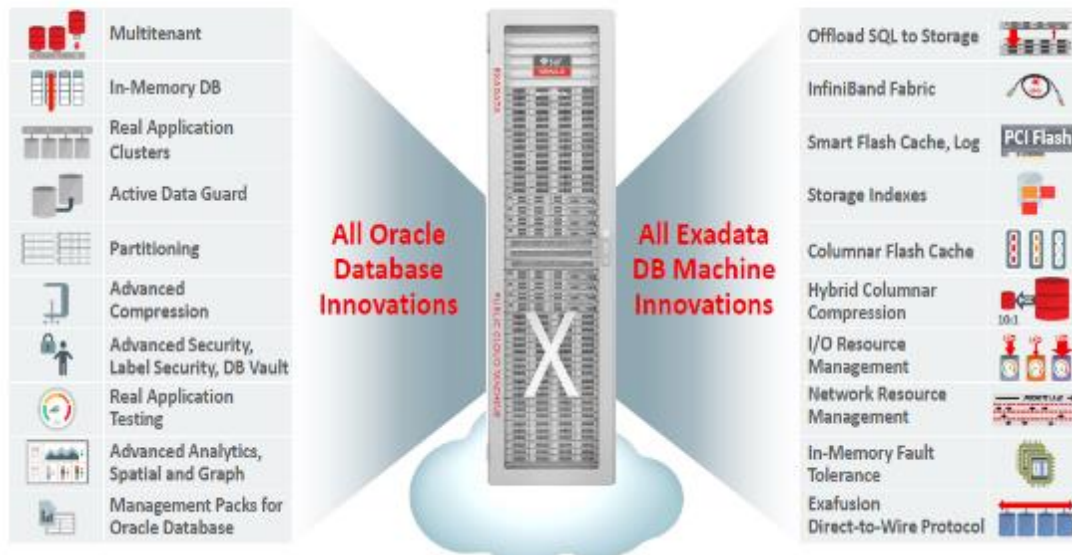
There are two types of IM expressions:

- user-defined In-Memory virtual column
- automatically detected IM expressions created by Expressions Statistics Store (ESS) which monitors workloads and store it as a hidden virtual column when captured :
 - DBMS_INMEMORY.IME_CAPTURE to capture the repeating expressions
 - DBMS_INMEMORY.IME_POPULATE to create the in-memory virtual columns.



Walgreens roadmap to use In-Memory features

Exadata Cloud Machine: Compatible, Scalable, Available
Decades of Database Innovation Proven at Millions of Mission-Critical Deployments



Consolidation of Oracle DBs on Exadata Cloud Machine:

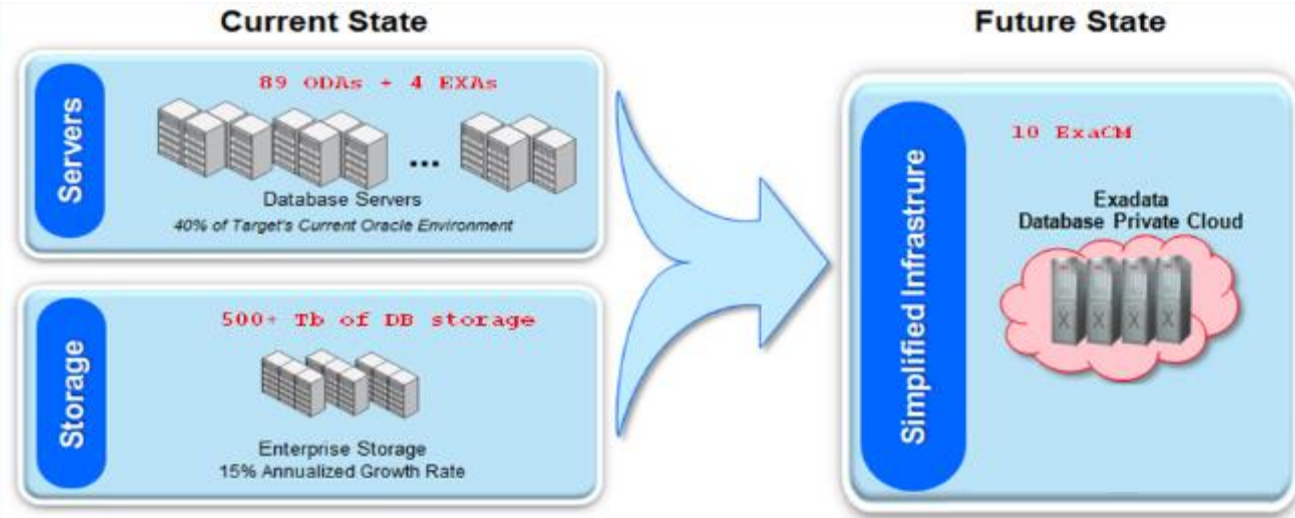
Taking advantage of Large Memory (720 Gb per node)

Taking advantage of all DB options included (in-Memory, Multitenant, Active Data Guard)

Taking advantage of fast storage and HCC compression options



Oracle DB consolidation on Exadata Cloud Machine platform



Significantly higher DB Consolidation Density and Datacenter Footprint (Multiple VMs with multi-tenant DB consolidation per network zone/environment/business unit)

3x reduction in overall power usage (14.5 kVA vs 44.8 kVA)

Spare on-demand capacity to migrate remaining ODA/Exadatas in the future or provision DBs for new applications almost instantaneously.



Acknowledgement

- ❖ Special Thanks to director Jim Hope and manager Glenn Campbell for all the support.
- ❖ Special Thanks to senior DBA Karen Moe who participated in the In-Memory POC in her system and provided feedback to us.
- ❖ Special Thanks to senior DBA Raghav Kalakota for setting up the POC environment in ODA.

References

- IOUG SELECT: Improving Performance With 12c In-Memory Option <http://select.ioug.org/blog/improving-performance-with-12c-in-memory-option> by Fong Zhuang
- Oracle White Paper: ORACLE DATABASE 12 C IN-MEMORY OPTION - The Top Tier of a Multi-tiered Database Architecture
- Oracle blogs: https://blogs.oracle.com/In-Memory/entry/what_is_an_in_memory
- Oracle White Paper: When to Use Oracle Database In-Memory
- Oracle white paper: Oracle Database In-Memory Advisor
- Oracle white paper: Oracle Database In-Memory
- IOUG presentation by Kai Yu: Optimize OLAP & Business Analytics Performance with Oracle 12c In-Memory Database Option
- Oracle Database In-Memory *Powering the Real-Time Enterprise*

Questions

