

Oracle Database In-Memory

Enabling Real-Time Analytics

Andy Rivenes
Database In-Memory Product Management
Oracle Corporation

Email: andy.rivenes@oracle.com
Twitter: @TheInMemoryGuy
Blog: blogs.oracle.com/in-memory



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Database Platform of the **Future** – Complete and *Integrated*

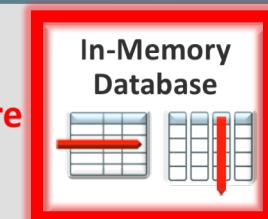
Services

Any Application
Any Data



Architecture

Fast, Low Cost, Secure
Scalable, Available



In-Database Multitenancy



In-Database Security



Fault-Tolerant Scale-Out



Systems

Engineered
for Databases



Cloud

Autonomous, Elastic
Wherever You Want



Public Cloud



Cloud At Customer

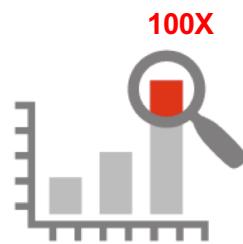


What is Database In-Memory



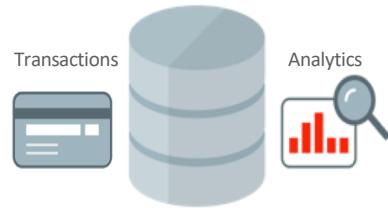
Oracle Database In-Memory

Real-Time Analytics



Enable Real-Time Business Decisions

Accelerate Mixed Workload



Run analytics on Operational Systems

Risk-Free



Proven Scale-Out, Availability, Security

Trivial to Implement



No Application Changes
Not Limited by Memory

Row Format Databases vs. Column Format Databases

Rows Stored Contiguously



- **Transactions** run faster on row format
 - Example: Query or Insert a sales order
 - Fast processing few rows, many columns

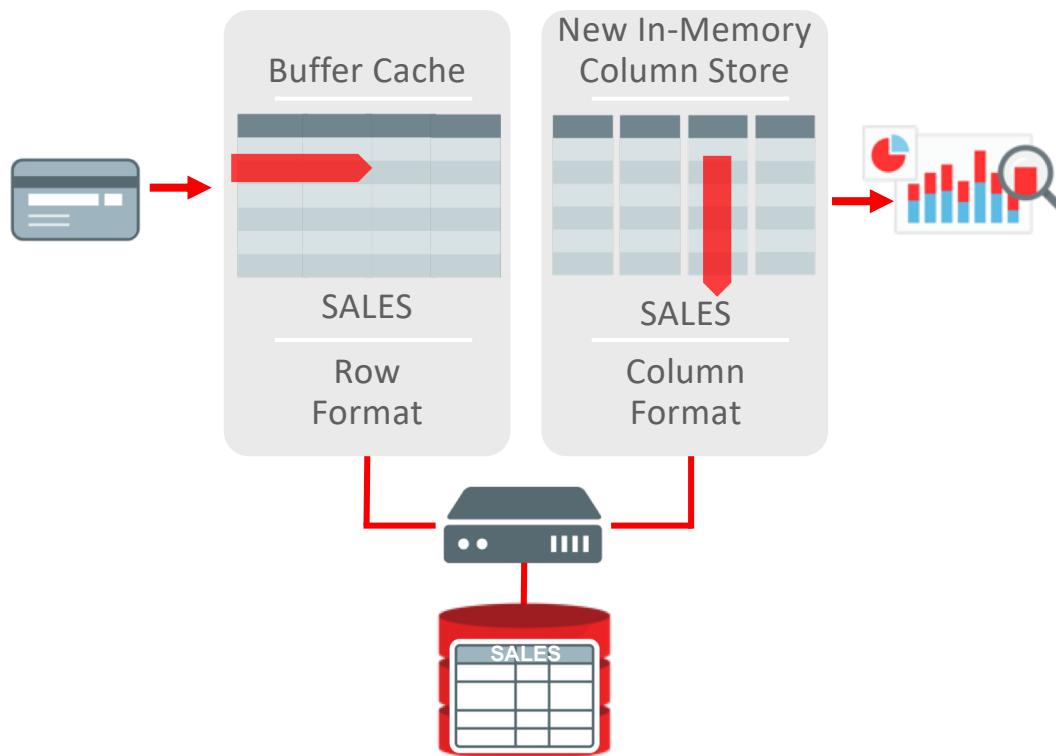
Columns Stored Contiguously



- **Analytics** run faster on column format
 - Example : Report on sales totals by region
 - Fast accessing few columns, many rows

Until Now Must Choose One Format and Suffer Tradeoffs

Breakthrough: Dual Format Database



- **BOTH** row and column formats for same table
- Simultaneously active and transactionally consistent
- Analytics & reporting use new in-memory Column format
- OLTP uses proven row format

The Forrester Wave™: In-Memory Databases, Q1 2017

- Oracle In-Memory Scored Highest of All Databases for Current Offering and Strategy
- For the full report see:
<https://reprints.forrester.com/#/assets/2/132/%27RES132143%27/reports>

Where Is It Available



Where Database In-Memory Is Available

- Database In-Memory is an option for Oracle Database Enterprise Edition
- Database In-Memory was included in the first patchset (12.1.0.2) for 12.1 and all subsequent Oracle Database releases
- Available:
 - Exadata Express Cloud Service – X1000IM – up to 10GB Column Store
 - Database Cloud Service: **Extreme Performance**
 - Exadata Cloud Service
 - Exadata Cloud at Customer
 - On-premises

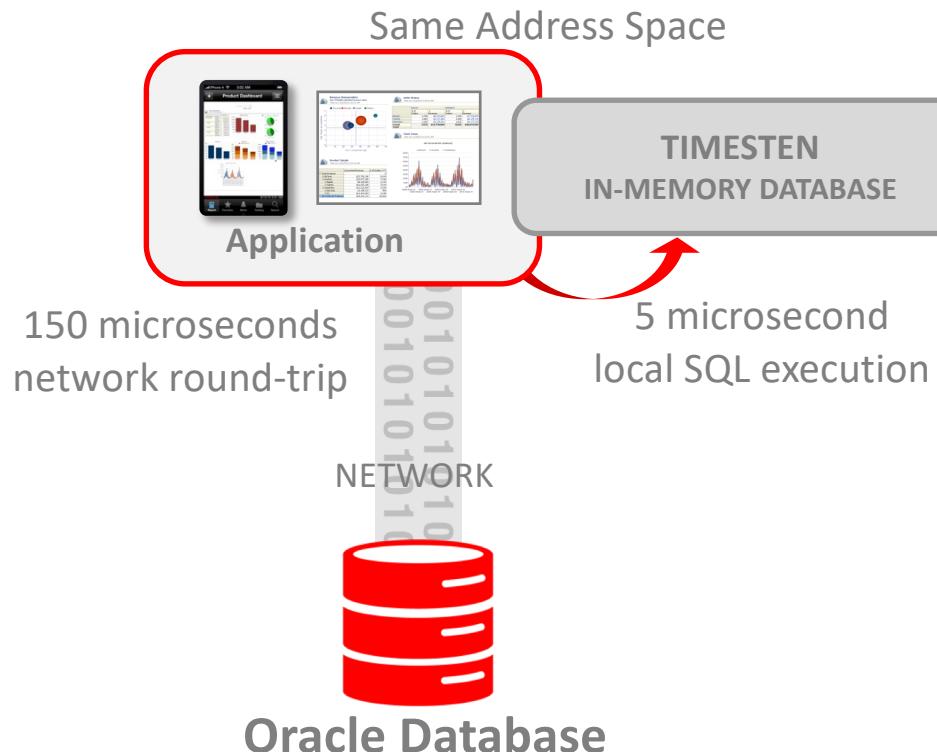


Note: Database In-Memory is **not** enabled by default

Isn't it just TimesTen



TimesTen for Latency-Critical OLTP Complementary In-Memory Technology



- **Latency-Critical OLTP limited by network between application and database**
 - Phone call routing, stock trading
- TimesTen In-Memory Database is light-weight and ultra-fast
 - Runs in application address space: **No Network**
 - **30x** faster latency-critical OLTP

How easy is it to get started



Oracle In-Memory: Simple to Implement

1. Configure Memory Capacity

- `inmemory_size = XXX GB`

2. Configure tables or partitions to be in memory

- `alter table | partition ... inmemory;`

3. Later drop analytic indexes to speed up OLTP

Oracle In-Memory Advisor

Object Type	Object	Estimated In-Memory Size	Analytics Processing Seconds	Estimated Reduced Analytics Processing Seconds	Estimated Analytics Processing Performance Improvement Factor	Benefit / Cost Ratio (Improvement Factor / In-Memory Size)
Table	SOE.LOGON	451.76MB	2114	1,887	9.3X	20.586
Table	SOE.CARD_DETAILS	607.32MB	8346	7,248	7.6X	12.514
Table	SOE.ADDRESSES	1.09GB	5237	4,621	8.5X	7.798
Partition	SOE.PRODUCT_MOCKUP.Y2014Q1	812.6MB	2003	1,489	3.9X	4.799
Table	SOE.CUSTOMERS	1.10GB	108	95	8.2X	7.455
Table	SOE.ORDER_ITEMS	2.19GB	7128	6,393	9.7X	4.429
Table	SOE.ORDERS	1.34GB	3512	2,917	5.9X	4.403
Table	SOE.PRODUCT_INFORMATION	1.78MB	2873	2,205	4.3X	2.416
Partition	SOE.PRODUCT_MOCKUP.Y2013Q4	1.62GB	97	1,489	3.7X	2.284
Partition	SOE.PRODUCT_MOCKUP.Y2014Q2	3.37GB	642	493	4.3X	1.276

- New In-Memory Advisor
- Analyzes existing DB workload via AWR & ASH repositories
- Provides list of objects that would benefit most from being populated into IM column store



Note: Database Tuning Pack license required

Oracle Compression Advisor And In-Memory

```
DECLARE
    l_blkcnt_cmp          PLS_INTEGER;
    l_blkcnt_uncmp         PLS_INTEGER;
    l_row_cmp              PLS_INTEGER;
    l_row_uncmp             PLS_INTEGER;
    cmp_ratio               PLS_INTEGER;
    l_comptype_str          VARCHAR2(100);
    comp_ratio_allrows      NUMBER := -1;

BEGIN
    dbms_compression.Get_compression_ratio (
        scratchtbsname => 'TS_DATA',
        ownname           => 'SSB',
        objname           => 'LINEORDER',
        subobjname        => NULL,
        comptype          => dbms_compression.comp_inmemory_query_low,
        blkcnt_cmp          => l_blkcnt_cmp,
        blkcnt_uncmp         => l_blkcnt_uncmp,
        row_cmp              => l_row_cmp,
        row_uncmp             => l_row_uncmp,
        cmp_ratio             => cmp_ratio,
        comptype_str          => l_comptype_str,
        subset_numrows        => dbms_compression.comp_ratio_allrows);
    dbms_output.Put_line('The IM compression ratio is '|| cmp_ratio);

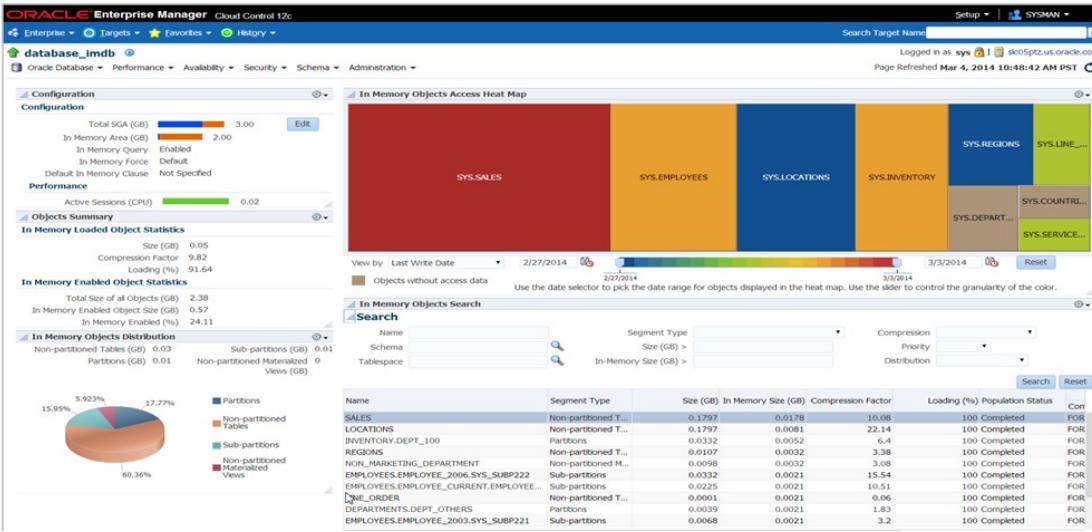
END;
/
```

- Easy way to determine memory requirements
- Use DBMS_COMPRESSION
- Applies MEMCOMPRESS to sample set of data from a table
- Returns estimated compression ratio



Oracle Enterprise Manager: In-Memory Central

In-Memory Central

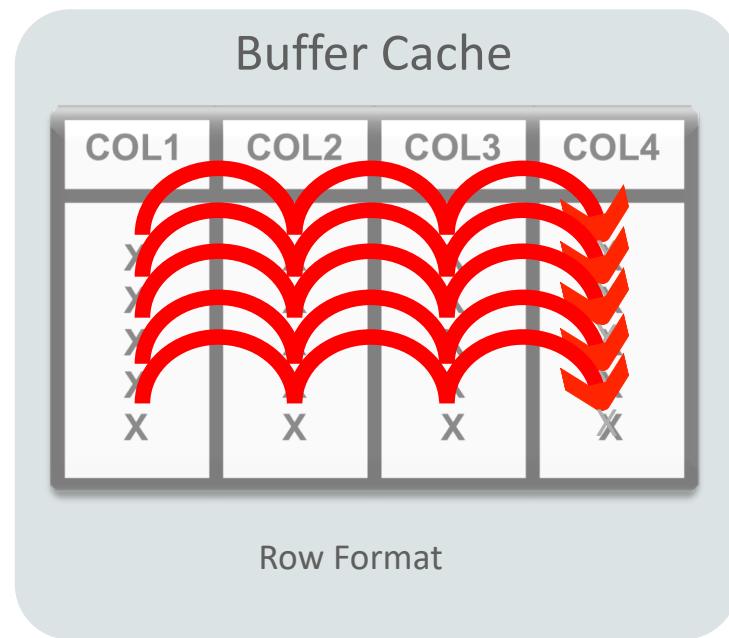


- OEM supports Database In-Memory
- In-Memory Central page gives a dashboard look to the IM column store
- Provides list of objects populated in the IM column store

How does it work



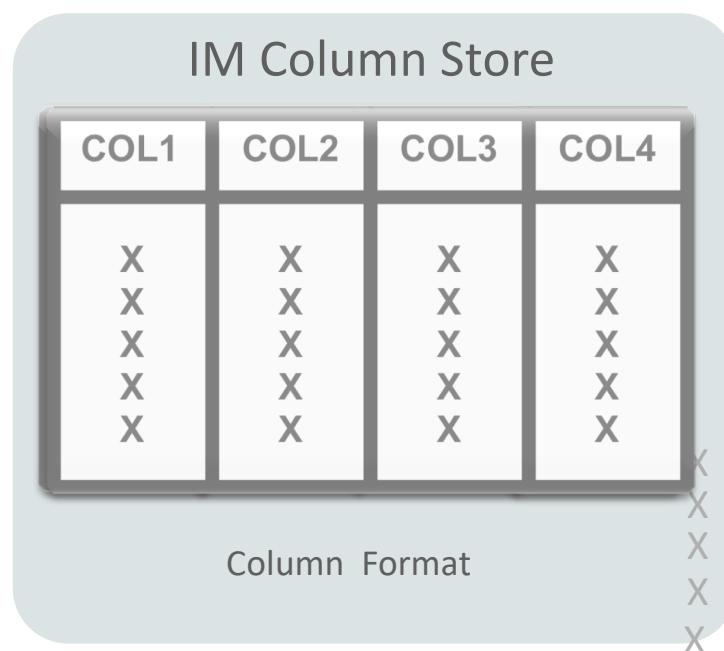
Why is an In-Memory scan faster than the buffer cache?



SELECT **COL4** FROM MYTABLE;



Why is an In-Memory scan faster than the buffer cache?

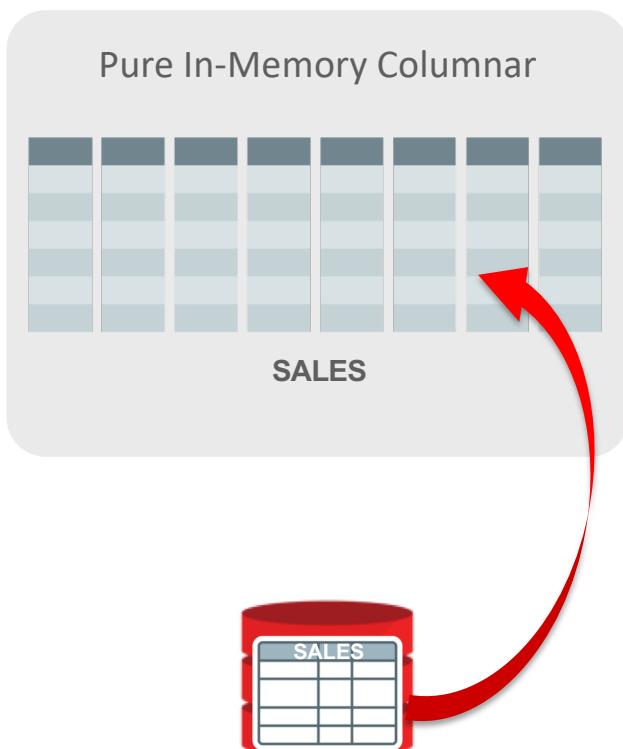


SELECT **COL4** FROM MYTABLE;



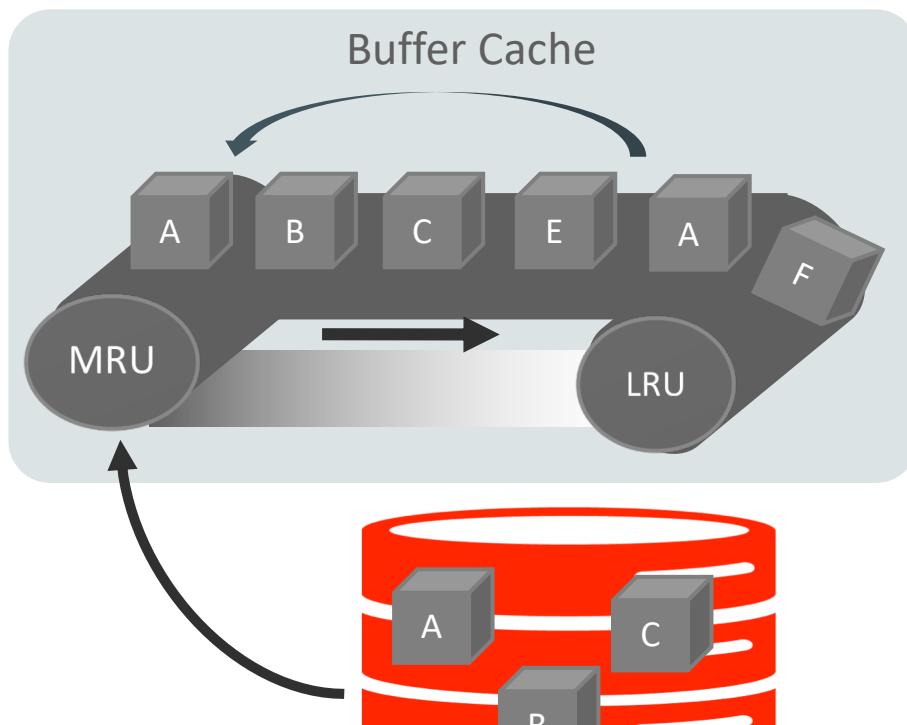
RESULT

Oracle In-Memory Columnar Technology



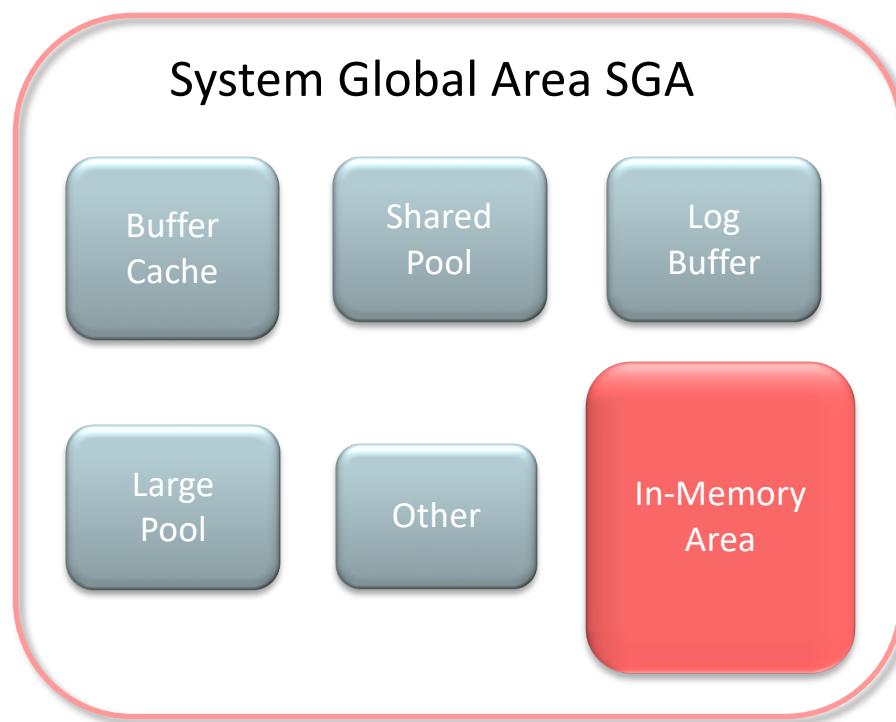
- Pure in-memory columnar format
 - Not persistent, and no logging
 - Quick to change data: fast OLTP
- Enabled at table or partition
 - Only active data in-memory
- 2x to 20x compression typical
- Available on all hardware platforms

In-Memory A Store – Not A Cache



- What is a cache?
- A pool of memory
- Data automatically brought into memory based on access
- Data automatically aged out
- Good example:
Oracle Database Buffer Cache

In-Memory Area: New Static Area within SGA



- Contains data in the new In-Memory Column Format
- Controlled by INMEMORY_SIZE parameter
 - Minimum size of 100MB
- SGA_TARGET must be large enough to accommodate this area



How Do I Get Data In And Out Of The In-Memory Column Store?

Populating : Enable Objects for In-Memory

```
ALTER TABLE sales INMEMORY;
```

```
ALTER TABLE sales NO INMEMORY;
```

```
CREATE TABLE customers .....
```

```
PARTITION BY LIST  
(PARTITION p1 ..... INMEMORY,  
(PARTITION p2 ..... NO INMEMORY) ;
```

- New INMEMORY ATTRIBUTE

- Eligible segment types are

- Tables
- Partitions
- Subpartitions
- Materialized views

- Following types not eligible

- IOTs
- Hash clusters

Pure OLTP
Features

- Out of line LOBs

Populating : Columns Can Be Excluded

```
ALTER TABLE sales INMEMORY
NO INMEMORY (delivery_note);
```

- You don't have to populate all columns
- It is possible to populate only certain columns
- Two phase approach
 1. INMEMORY attribute on Table automatically inherited by columns
 2. Need to remove attribute from the columns you don't want populated

Why not just “cache”
the table in the row
store



Compare Column-store to Row-store

```
SQL> -- In-Memory Column Store query
SQL>
SQL> select max(lo_ordtotalprice) most_expensive_order From LINEORDER;

MOST_EXPENSIVE_ORDER
-----
57346348

Elapsed: 00:00:00.01

| Id | Operation           | Name      | Rows | Bytes | Cost (%CPU)| Time   |
| 0  | SELECT STATEMENT    |           |       |        | 5401 (100) |          |
| 1  | SORT AGGREGATE     |           |     1 |       6 |            |          |
| 2  | TABLE ACCESS INMEMORY FULL| LINEORDER | 59M  | 343M  | 5401  (16) | 00:00:01 |

SQL> -- Buffer Cache query with the column store disabled via NO_INMEMORY hint
SQL>
SQL> select /*+ NO_INMEMORY */ max(lo_ordtotalprice) most_expensive_order From LINEORDER;

MOST_EXPENSIVE_ORDER
-----
57346348

Elapsed: 00:00:08.38

| Id | Operation           | Name      | Rows | Bytes | Cost (%CPU)| Time   |
| 0  | SELECT STATEMENT    |           |       |        | 123K(100) |          |
| 1  | SORT AGGREGATE     |           |     1 |       6 |            |          |
| 2  | TABLE ACCESS FULL  | LINEORDER | 59M  | 343M  | 123K  (1)  | 00:00:05 |
```



Why Are Analytic Queries Faster In The In-Memory Column Store?

Database In-Memory Technology

Scanning and filtering data more efficiently

Columnar Format



Access only the columns you need

Compression



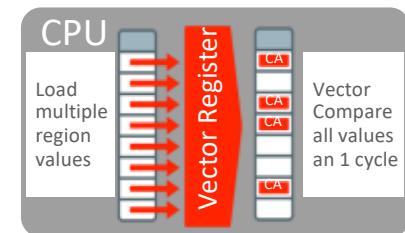
Scan & filter data in compressed format

Storage Indexes



Prune out any unnecessary data from the column

SIMD Vector Processing



Process multiple column values in a single CPU instruction

Optimizer Enhancements

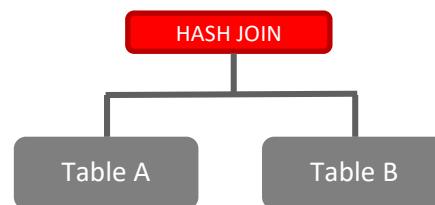
Improves all aspects of analytic queries

Data Scans



- Speed of memory
- Scan and Filter only the needed Columns
- Vector Instructions

Joins



- Convert Star Joins into 10X Faster Column Scans
- Search large table for values that match small table

In-Memory Aggregation



- Create In-Memory Report Outline that is Populated during Fast Scan
- Runs Reports Instantly

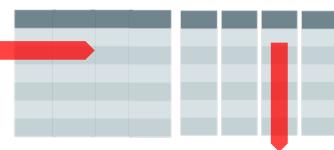
Improvements in 12.2 for Database In-Memory

Real-Time Analytics



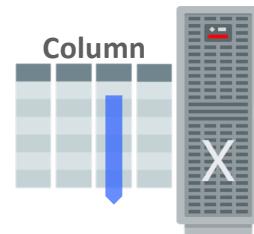
- **2X** Faster Joins
- **5X** Faster Expressions

Mixed Workload



- Active Data Guard Support

Massive Capacity



- In-Memory on Exadata Flash (HCC Objects)

Multi-model



- Native support for JSON Data type

Automation



- Dynamic Data Movement Between Storage & Memory
- IM FastStart
- IM Column Store Re-sizing

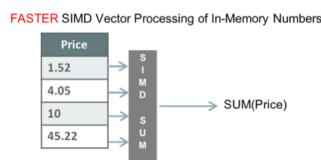
Improvements in 18c for Database In-Memory

Further Performance Gains



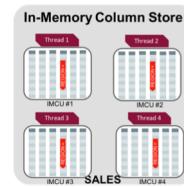
2X Query Performance Gains
Exadata Flash support for non-HCC objects

Optimized Arithmetic



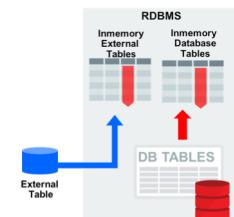
In-Memory Optimized Arithmetic

Dynamic Scans



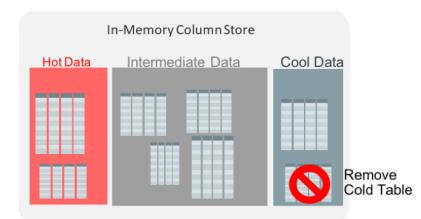
In-Memory Dynamic Scans

External Tables



In-Memory External Tables

Automatic In-Memory

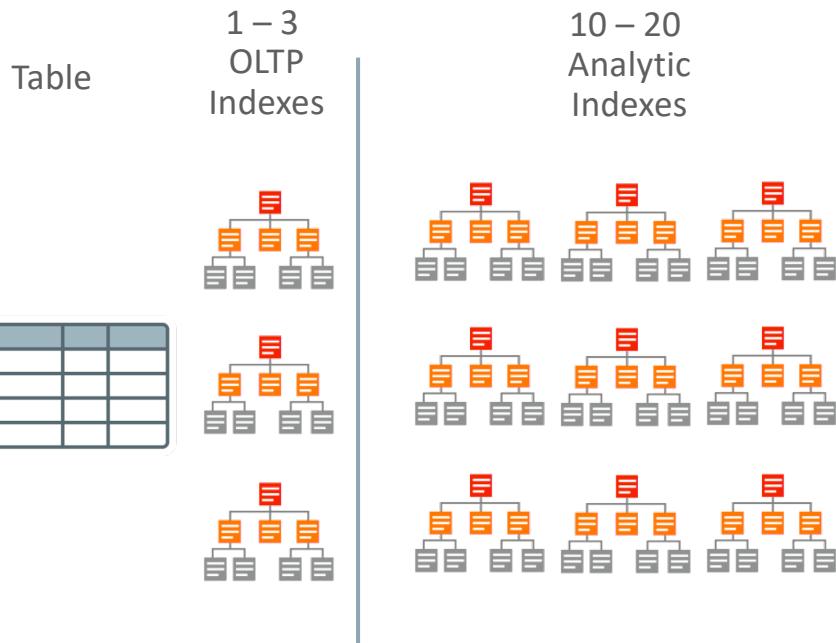


Automatic Data Movement Between Storage & Memory

How does it impact OLTP environments



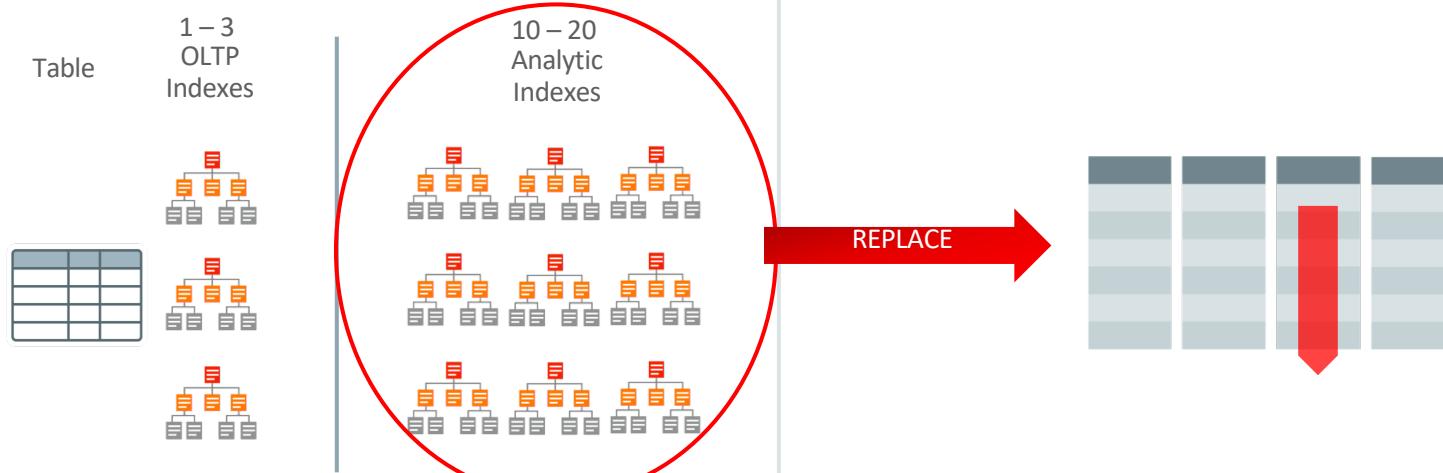
Complex OLTP is Slowed by Analytic Indexes



- Most Indexes in complex OLTP (e.g. ERP) databases are only used for analytic queries
- Inserting one row into a table requires updating 10-20 analytic indexes: **Slow!**
- Indexes only speed up predictable queries & reports

Database In-Memory Accelerates Mixed Workloads

- Complex OLTP is Slowed by Analytic Indexes



- Inserting one row into a table requires updating 10-20 analytic indexes: **Slow!**

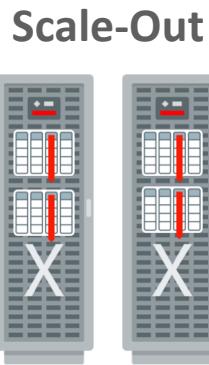
- Column Store Replaces Analytic Indexes

- Fast analytics on any columns
- Column Store not persistent so update cost is much lower

Database In-Memory and other Oracle Database features



Database In-Memory: Scale Out



- Scale-Out Across Servers to Grow Memory and CPUs
- In-Memory Queries Parallelized Across Servers

Scale-Up



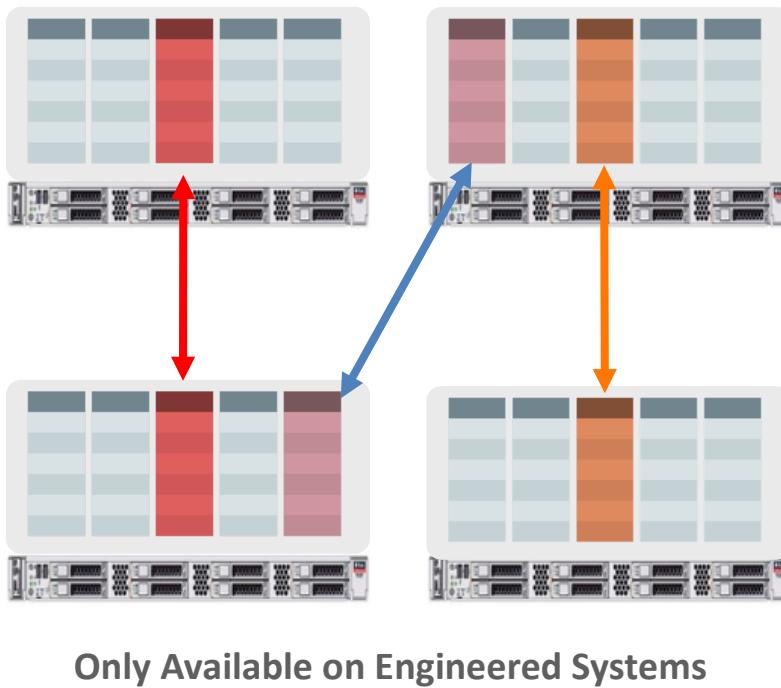
- Scale-Up on large SMPs
- NUMA Optimized

Combine with Flash and Disk



- Easily place data on most cost effective tier
- Simultaneously Achieve:
 - Speed of DRAM
 - I/Os of Flash
 - Cost of Disk

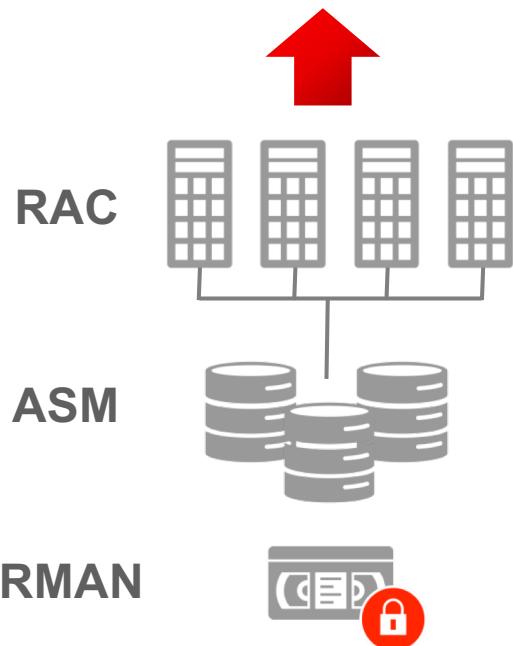
Database In-Memory: Unique Fault Tolerance



- Similar to storage mirroring
- Duplicate in-memory columns on another node
 - Enabled per table/partition
 - e.g. only recent data
 - Application transparent
- Downtime eliminated by using duplicate after failure

Database In-Memory: Industrial Strength Availability

Data Guard & GoldenGate



- Pure In-Memory format does not change Oracle's storage format, logging, backup, recovery, etc.
- All Oracle's proven availability technologies work transparently
- **Protection from all failures**
 - Node, site, corruption, human error, etc.

What Workloads Benefit From Database In-Memory



What is an analytic query?

Using Aggregation to Find Patterns or Trends in the Data

Which products give us our highest margins?

Who are the top 10 sales reps in the north west region this month?

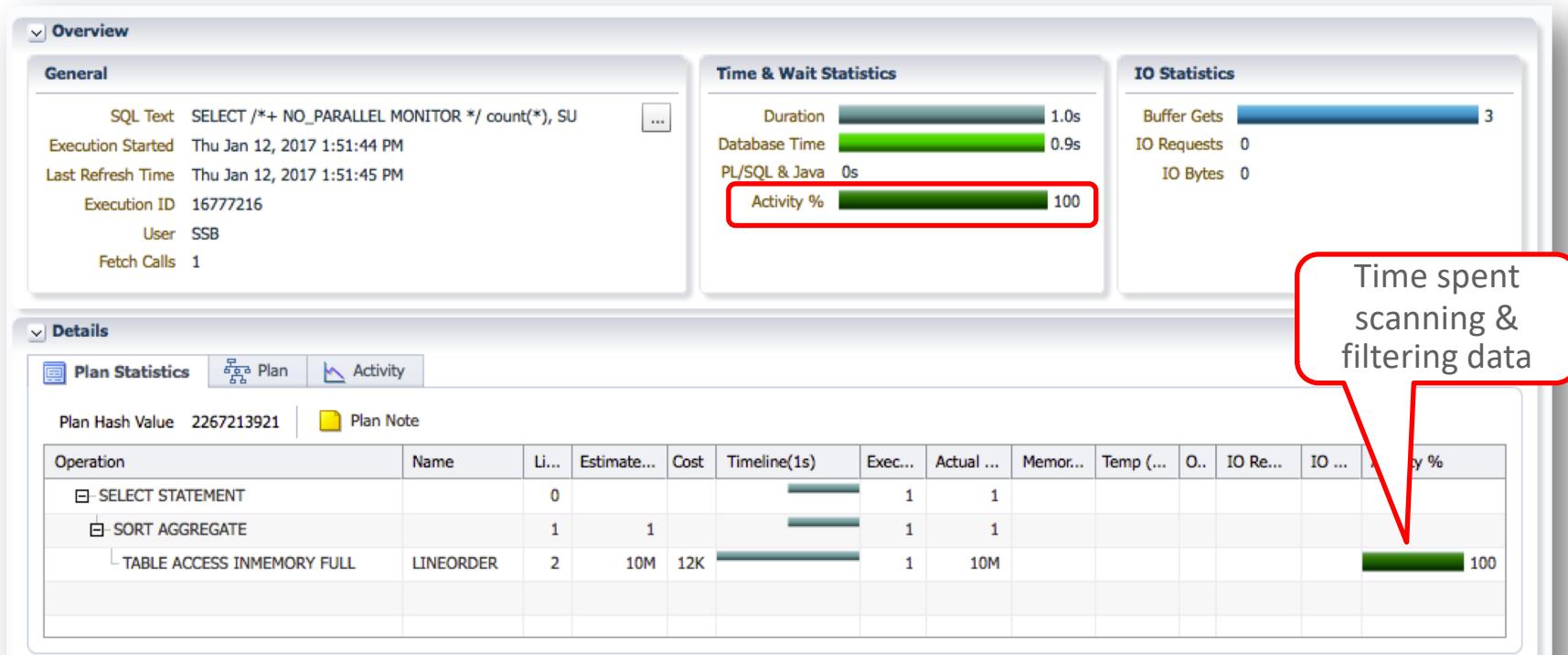
If I get a 20% discount on widget A, how much will our margins improve?

Queries that Benefit from Database In-Memory

Analytic Queries

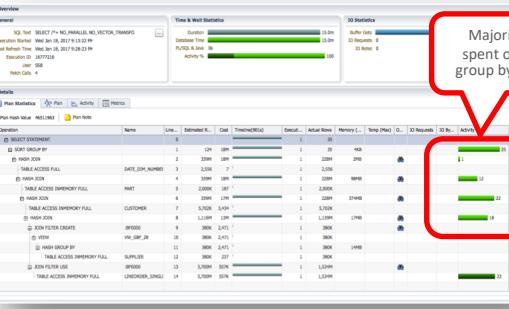
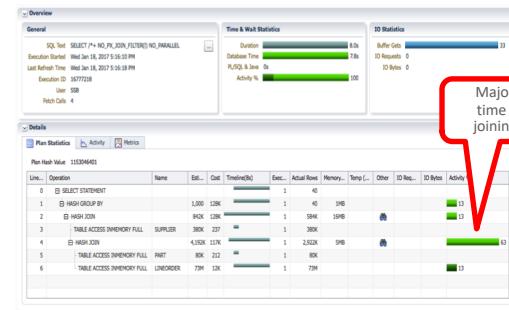
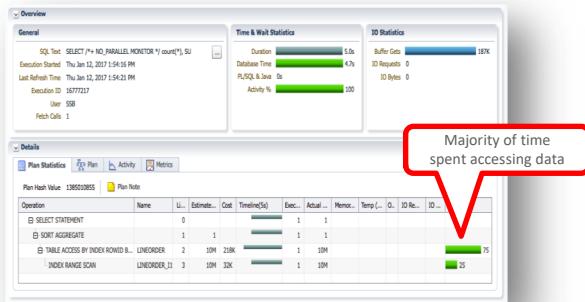
- Return as few rows as possible
- Limit the number of columns accessed
- Use aggregation to return fewer values
- Use selective column predicates
- Use selective join conditions
- Limit the number of tables being joined
- Avoid complex SQL functions

The Ideal In-Memory Query



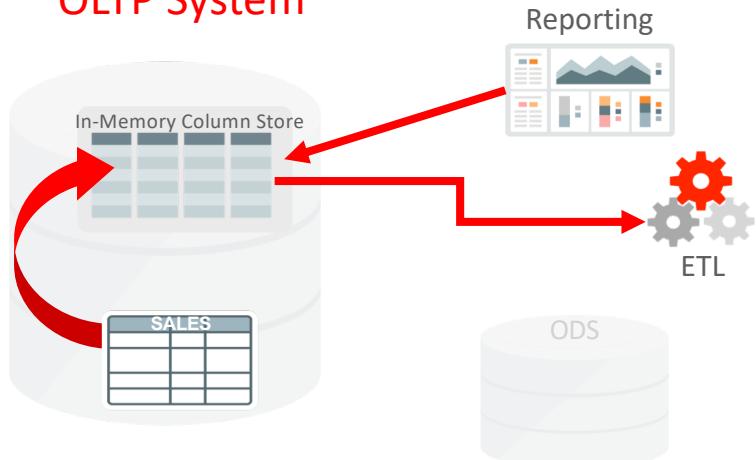
When Database In-Memory Helps

- For a non-trivial amount of rows and execution time:
 - Majority of time spent accessing data
 - Majority of time spent joining data
 - Majority of time spent on scan and group by operations



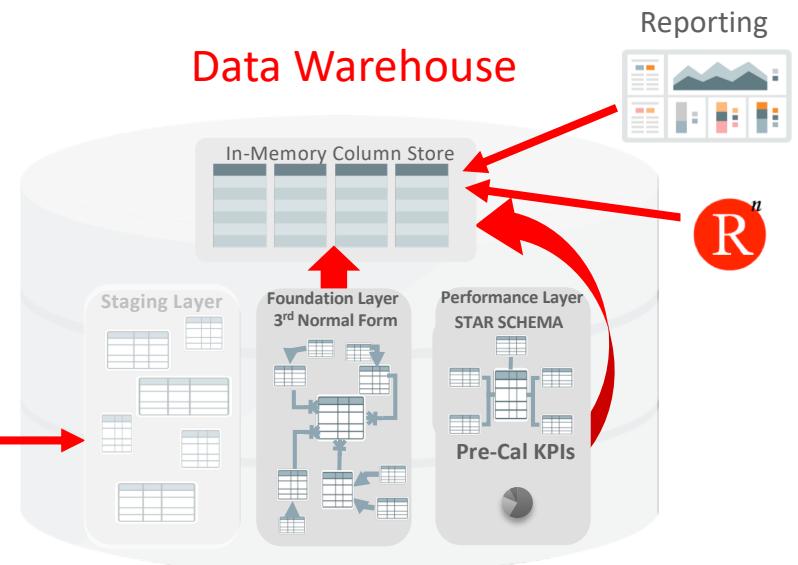
Where to use In-Memory

OLTP System



- Enables real-time reporting directly on OLTP data
- Speeds data extraction part of ETL process
- Removes need for separate ODS
- Speeds up mixed workload

Data Warehouse



- Star-schema and pre-calculated KPIs
 - Improves performance of dash-boards
- All or a subset of Foundation Layer
 - For time-sensitive analytics on 3rd normal form
- Staging/ETL/Temp not good candidates
 - Write once, read once

How are customers using Database In-Memory



Database In-Memory References

AT&T WiFi – Data Warehouse



- Business Objects reports **100X** faster
- ETL processes improved by **50%** faster
- No changes to SAP Business Objects reports

Villeroy & Boch – SAP BW



- SAP BW COPA queries **30 – 33X** faster
- SAP Transaction list queries **4 – 4,800X** faster
- Avoided expensive & risky upgrade to S4/Hana

BOSCH – SAP CRM



- **Dropped** all custom indexes
- Analytic queries **2-20X** faster, DML **2-3X** faster
- No changes to application required

Die Mobiliar – Mixed Workload



- Analytic queries **50-200X** faster
- Database size **reduced** considerably
- Phase out of Netezza and mainframe systems

Database In-Memory Customers

Mankind Pharma – Mixed Workload

- Analytical reports **11x** faster
- Dropping indexes improved OLTP
- **90% reduction** in database size



LION – SAP ERP

- Analytic queries **4X** faster
- Transactions **2X** faster
- Analytic queries now possible on 100s of Millions of Point-of-Sales Transactions



Shanghai Customs – Mixed Workload

- Processes Clearance **43x** Faster
- Improves Declaration-Services Efficiency
- Reduced Costs



Lufthansa – Reporting Application

- Analytic queries up to **100x** faster
- Improved data ingest performance
- Reduction in database size



Where can I get more information



Additional Resources



Join the Conversation

- https://twitter.com/db_inmemory
- <https://twitter.com/TheInMemoryGuy>
- <https://blogs.oracle.com/In-Memory/>
- <https://www.facebook.com/OracleDatabase>
- <http://www.oracle.com/goto/dbim.html>

ORACLE®

White Papers (otn.com)

- [Oracle Database In-Memory White Paper](#)
- [Oracle Database Implementation and Usage White Paper](#)
- [Oracle Database In-Memory Aggregation Paper](#)
- [When to use Oracle Database In-Memory](#)
- [Oracle Database In-Memory Advisor](#)

Videos

- [Oracle Database In-Memory YouTube Channel](#)
- [oracle.com](#)
[Powering the Real-Time Enterprise](#)
[oracle.com/us/corporate/events/dbim/index.html](#)
[Real-Time Analytics Demo](#)
- YouTube - Juan Loaiza: [DBIM: What's new in 12.2](#)

Additional Questions

- In-Memory blog: [blogs.oracle.com/In-Memory](#)
- My email: andy.rivenes@oracle.com