Utilizing Oracle In-Memory to improve Application Performance

Ahmed Alomari

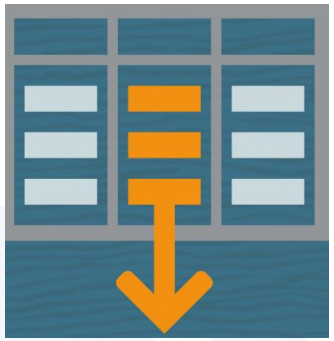
aalomari@cybernoor.com

Agenda

- What is In-Memory
- Why In-Memory
- Application Use Cases
- Q & A

What is In-Memory

What is In-Memory



- Data is stored in system memory to improve access performance.
  - Avoids latency of having to load/transform the data
  - Avoids physical I/O
  - Leverages compute layer to “divide-and-conquer”
- Optimizes access to data which is frequently referenced
  - Transactional data (i.e. dashboards/reporting)
  - Lookups/setup/reference data
- Expectation is orders of magnitude improvement

What is In-Memory

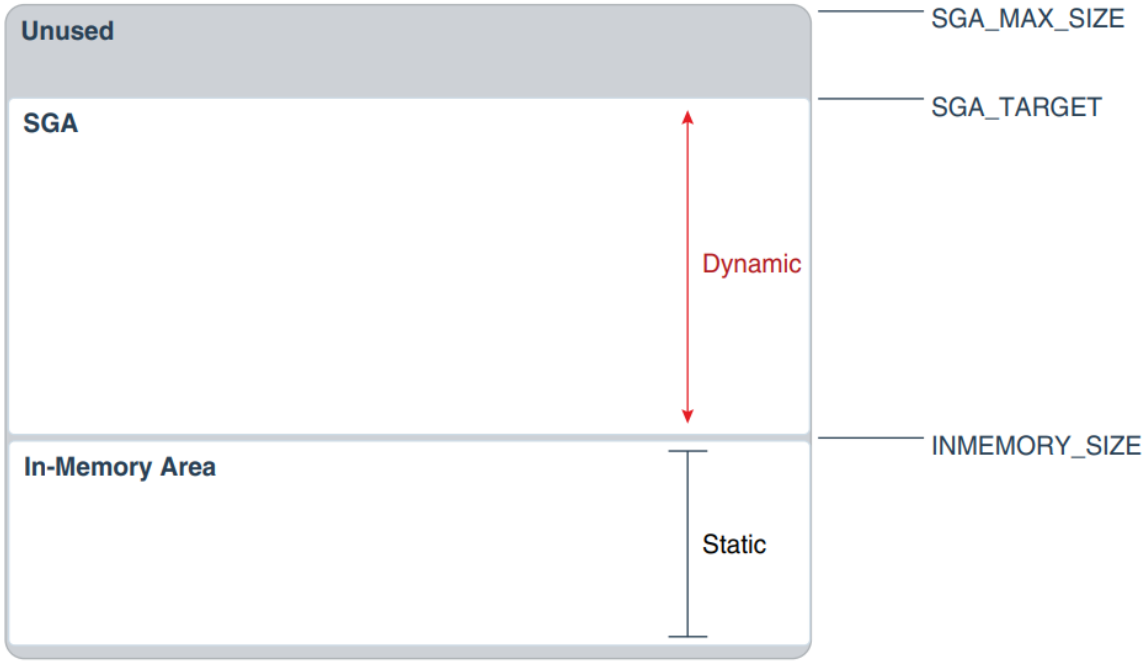
- Example: caching data using application caching to avoid SQL execution.

```
select *
into l_tax_rec
from AR_SYSTEM_PARAMETERS_ALL
where set_of_books_id = p_set_of_books;
```

- 33 seconds (500K executions) – SQL Lookup
- 40 ms (500K executions) – API getter
- Speed-up: 832X

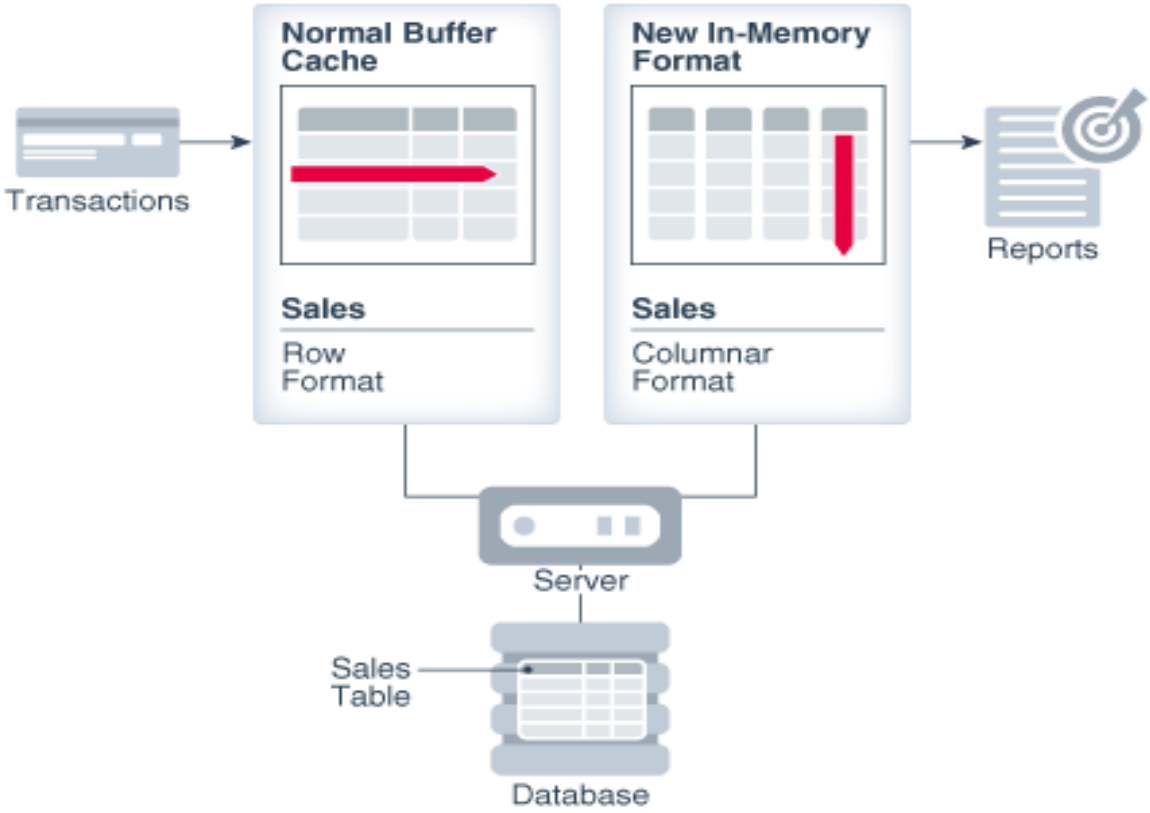
What is In-Memory

- Oracle Database In-Memory
  - IM column store – compressed columnar format
  - Part of the SGA
    - inmemory_size parameter



What is In-Memory

- Oracle Database In-Memory
  - IM column store – compressed columnar format



Why In-Memory?

Why In-Memory

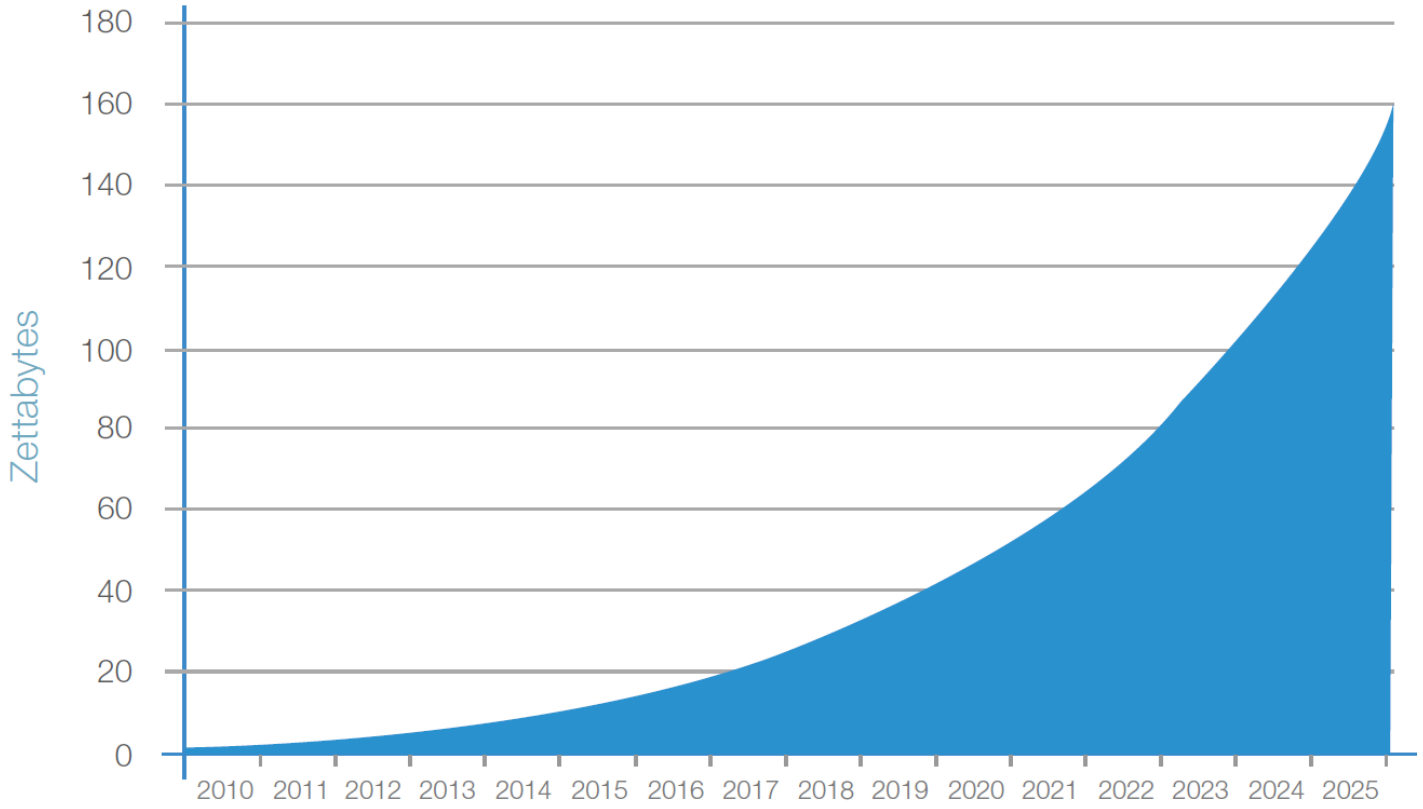
- Data is growing at phenomenal rates.
- Users require responsive performance for online transactions and searches.
- Business throughput depends on query performance.
- Provide data insight rather than just storage and traditional reporting.
- Data Analytics including AI and ML.
- Ingestion of large streams of data.

Why In-Memory

- Unstructured data will represent the bulk of the data footprint.
- Internet of things (IoT)
  - Sensors
  - Devices
  - Meters
  - Feeds
- Ability to search and analyze large streams of data is an application must ; not a nice to have.

Why In-Memory

- Data Growth Trends

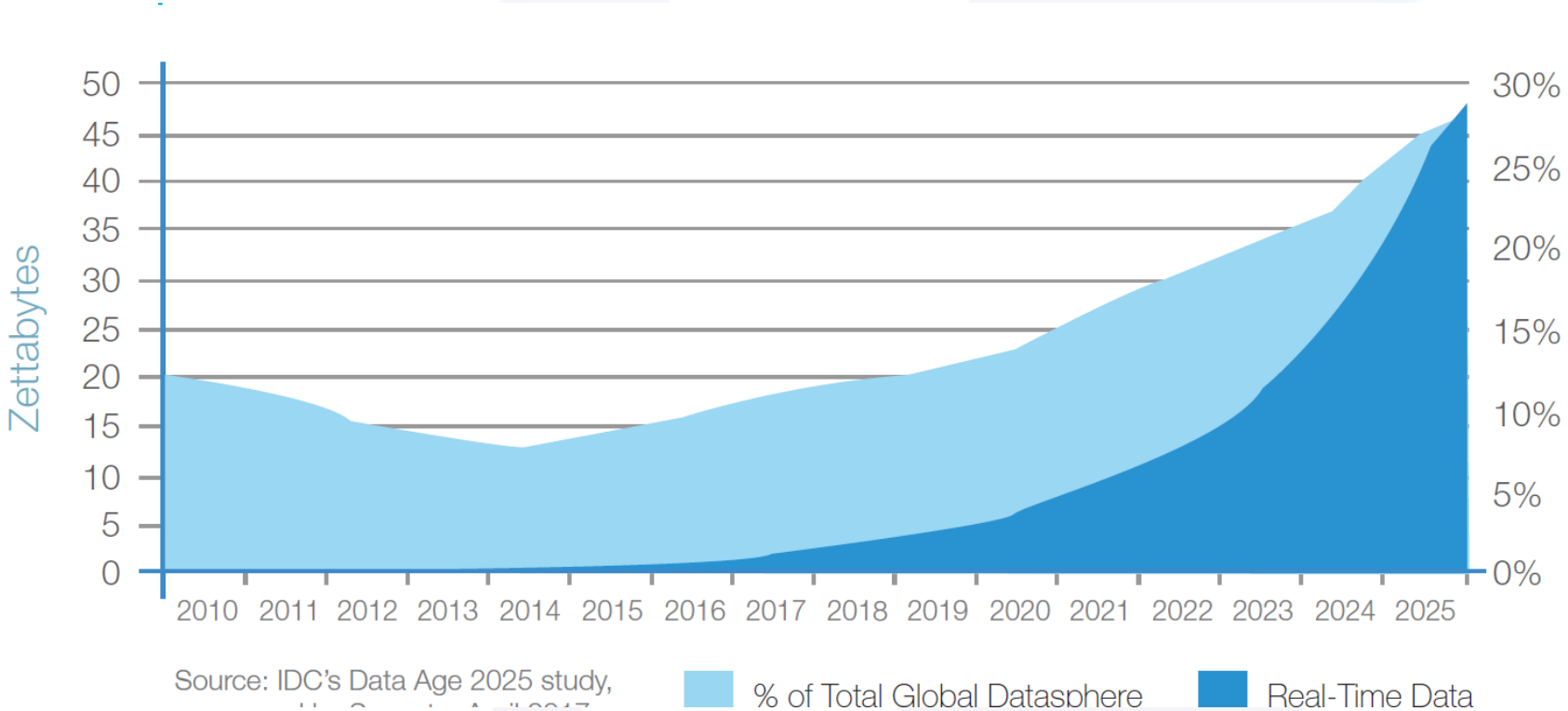


- Source: IDC's Data Age 2025 study

cybernoor™

Why In-Memory

- Data Growth Trends



- Source: IDC's Data Age 2025 study

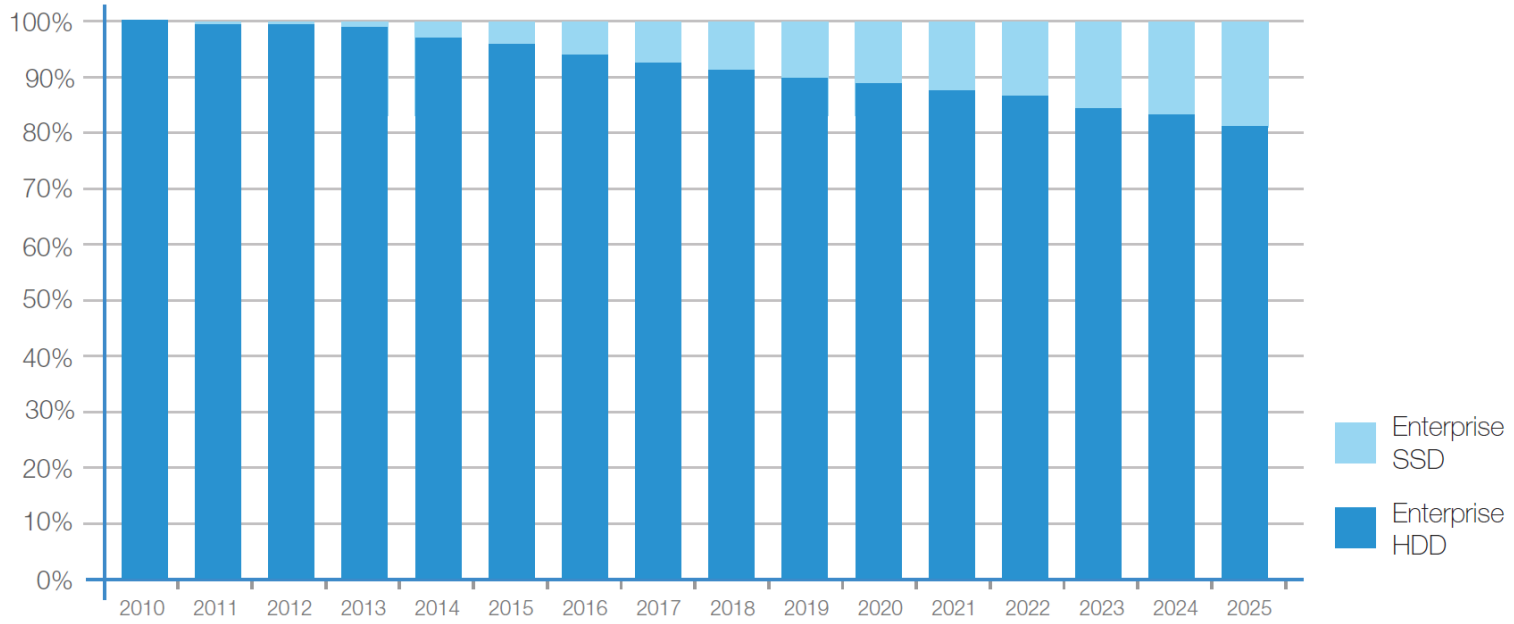
cybernoor™

Why In-Memory

- Data Storage Trends
  - Adoption of faster storage such as SSDs
  - Air / Helium drives
  - Larger capacity drives
    - 2007 (1 TB disk)
    - 2010 (2 TB disk)
    - 2020 (100 TB SSD)
    - 2025+ 50+ TB hard disk
  - NVMe (Nonvolatile Memory Express)
    - Latency 2-3 microseconds
    - 1 Million IOPS
    - Queue depth 64K
  - Cloud based Storage

Why In-Memory

- Data Storage Trends



- Source: IDC's Data Age 2025 study

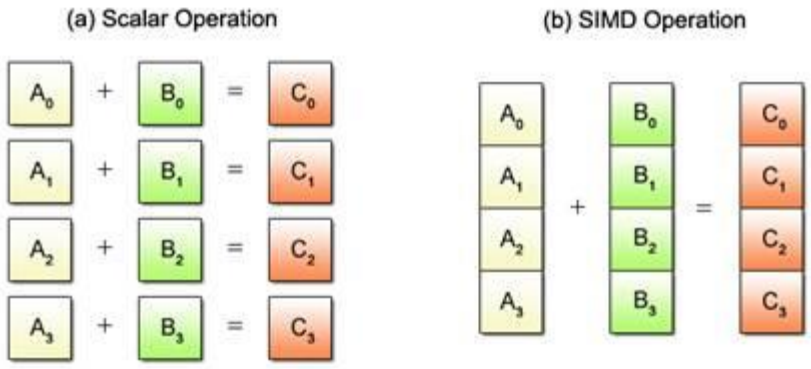
cybernoor™

Why In-Memory

- Building and maintaining application level data caches or disparate data lakes is time consuming and introduces logistical challenges
  - Cache invalidation
  - Data staleness
  - Data consistency issues when caches are updated
  - Data is replicated across application nodes
  - Application code itself has to be built to leverage such caches
  - Integration challenges
  - Data Security
  - Denorm tables/summary tables/MVs have to be built to match static set of queries.

Why In-Memory

- Oracle Database In-Memory allows developers to continue to utilize native SQL interfaces.
  - SIMD Vector Processing
    - Single Instruction Multiple Data



Why In-Memory

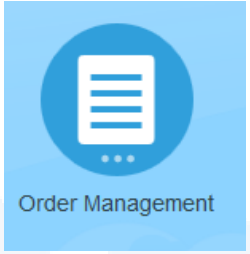
- Oracle Database In-Memory allows developers to continue to utilize native SQL interfaces.
  - Auto in-memory
    - INMEMORY_AUTOMATIC_LEVEL=HIGH (21c)
  - IM Storage Indexes
  - SIMD Vector Processing
  - JSON Documents
  - In-Memory Hybrid Scans (21c)
    - Useful for packaged applications where subsets of the data would be in the IM store and the rest in the buffer cache (row store).
  - External Tables

Why In-Memory

- Leverage native database security.
- Operate directly on the data source.
- Allows applications to achieve optimal performance for complex searches as well as data insights.
- Avoids having to build out data lakes in order to serve data analytics and/or complex reporting.
- In conjunction with RAC, allows the solution to scale out by expanding the cluster.
  - Leverage RAC services to attain application affinity.

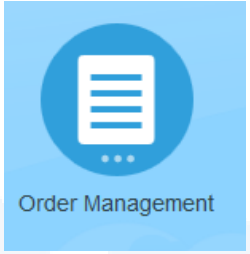
Application Use Cases

Order Management



- Improve performance of Order Searches as well as key batch cycles
  - Pick Release
  - Workflow Background Engine
  - Interface Trip Stop
  - Order Searches
  - Credit Summary Refresh
- Allows a decent number of secondary indexes on oe_order_headers_all and oe_order_lines_all to be eliminated.

Order Management

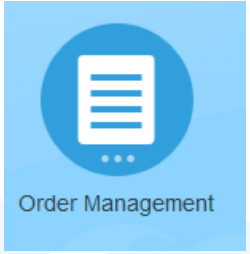


- Improve performance of Order Searches as well as key batch cycles

```
ALTER TABLE ONT.OE_ORDER_LINES_ALL INMEMORY PRIORITY CRITICAL;


SEGMENT_NAME       PRIORITY    COMPRESSION       SIZE_GB    INMEMORY_SIZE
OE_ORDER_LINES_ALL CRITICAL    FOR QUERY LOW      35.28        4.19
```

Order Management

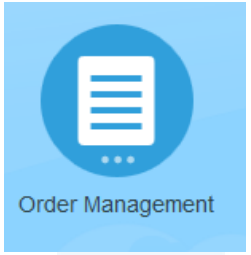


- Improve performance of Order Searches as well as key batch cycles

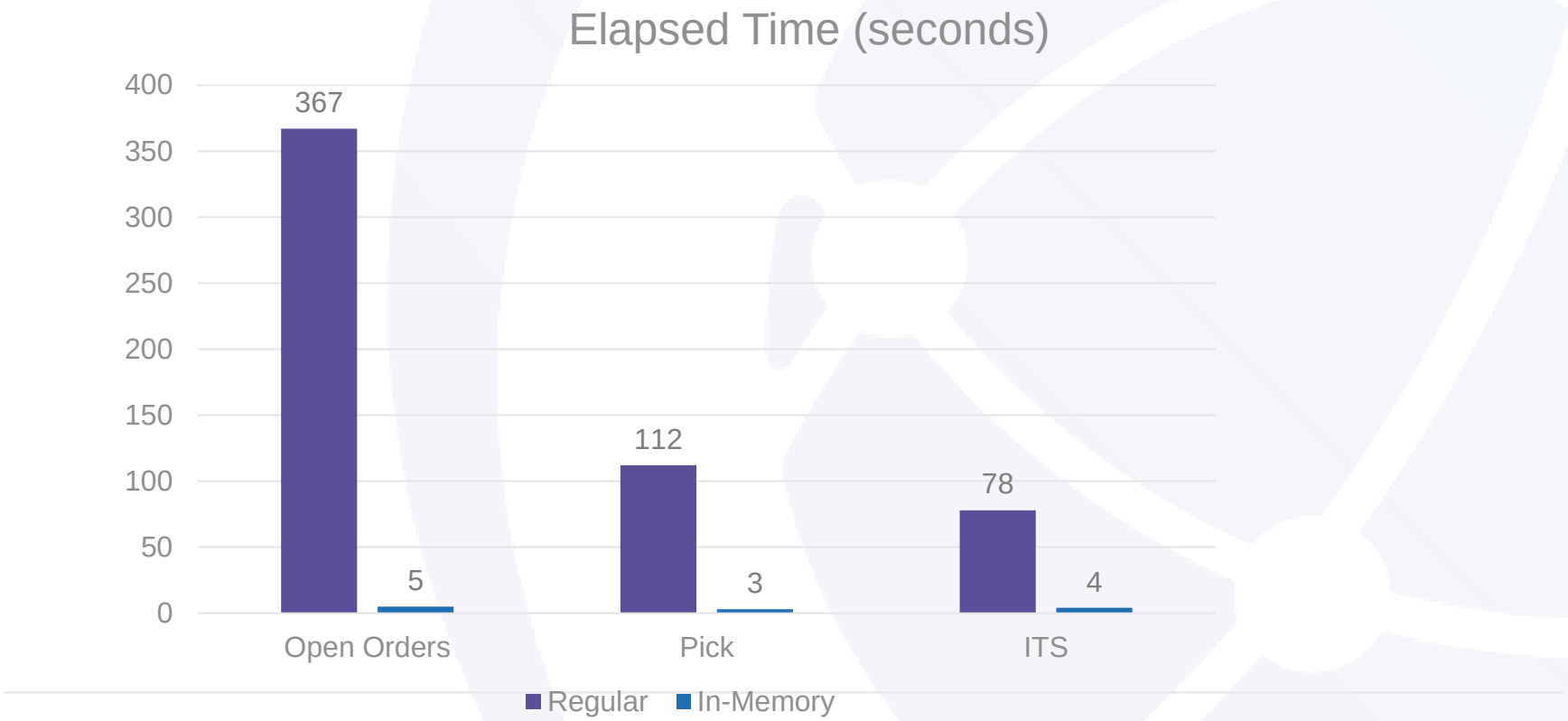
```
    SELECT ooh.org_id, wdd.delivery_detail_id, ooh.order_type_id, wdd.customer_id,
      wdd.organization_id, hz.party_name AS customer_name,
      hca.account_number AS customer_number, ool.header_id, ool.line_id,
      ooh.order_number, wdd.date_scheduled, wdd.date_requested,
      ool.line_number, hcsu.site_use_id, hcsu.LOCATION, wdd.ship_set_id,
      wdd.inventory_item_id, wdd.ship_from_location_id,
      . . . . . .
  FROM wsh_delivery_details wdd,
      wsh_delivery_assignments_v wda,
      oe_order_lines ool,
. . . . . .
. . . . . .
-----------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                     |       |       | 49434 (100)|          |
|   1 |  SORT ORDER BY               |                     |   874 |  8740 | 49434   (7)| 00:00:02 |
|   2 |   HASH GROUP BY              |                     |   874 |  8740 | 49434   (7)| 00:00:02 |
. . . . . .
. . . . . .
|* 23 |    TABLE ACCESS INMEMORY FULL| OE_ORDER_LINES_ALL | 45741 |   446K| 49430   (7)| 00:00:02 |
-----------------------------------------------------------------------------------------------
```

cybernoor™

Order Management

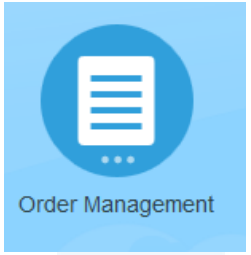


- Improve performance of Order Searches as well as key batch cycles

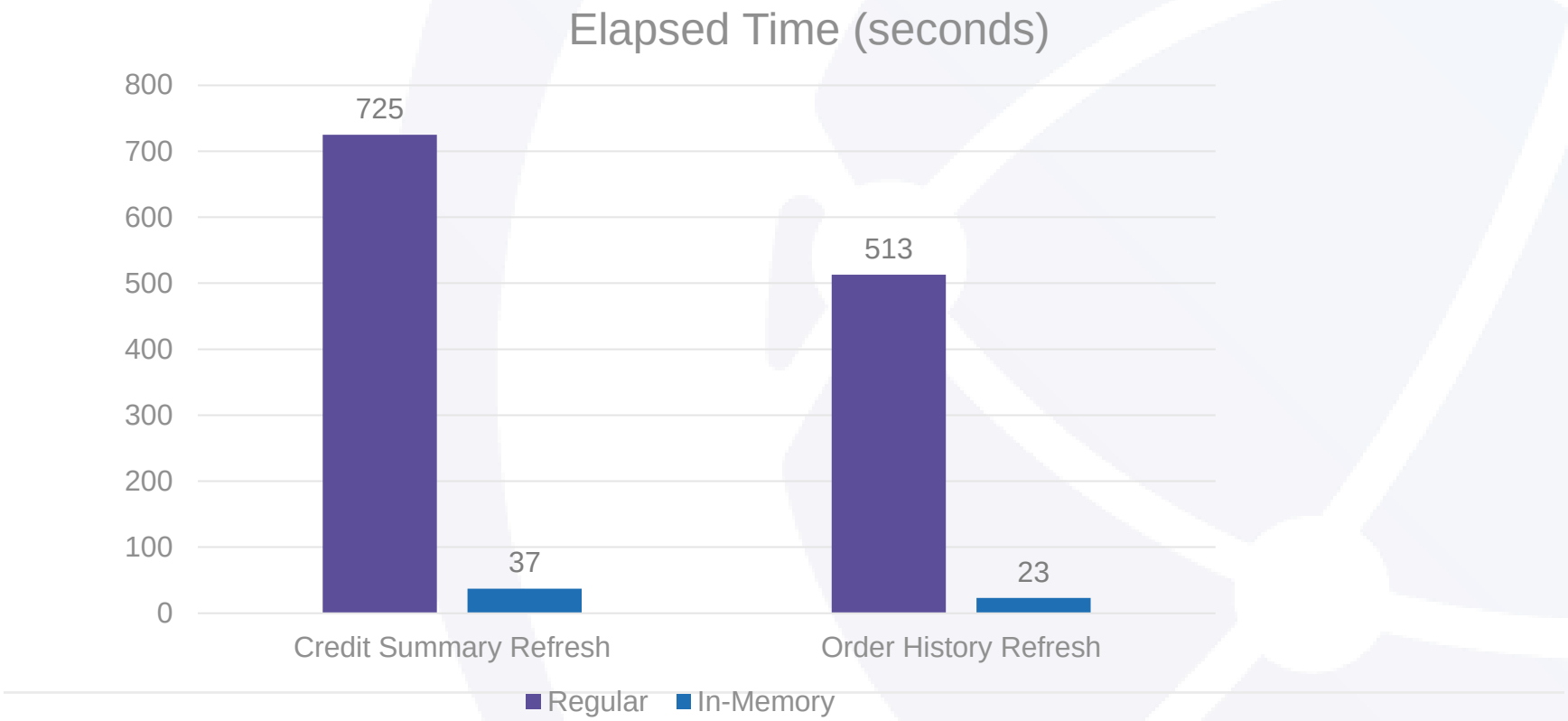


cybernoor™

Order Management



- Improve performance of Credit Summary



cybernoor™

Value Chain Planning (VCP)

- Improve performance of the planning cycle
  - Collections
  - Planning Data Pull
  - In-Memory Planner

```
ALTER TABLE MSC.MSC_SYSTEM_ITEMS INMEMORY PRIORITY CRITICAL;

ALTER TABLE MSC.MSC_BOM_COMPONENTS INMEMORY PRIORITY CRITICAL;

. . . . .

ALTER TABLE MSC.MSC_ROUTING_OPERATIONS INMEMORY PRIORITY
CRITICAL;
```

Value Chain Planning (VCP)

- Improve performance of the planning cycle
  - In-Memory Planner

```
87         87         87              HASH JOIN  (cr=5 pr=0 pw=0 time=567 us cost=3 size=903 card=43)
        87         87         87               INDEX RANGE SCAN MSC_PLAN_ORGANIZATIONS_U1 (cr=3 pr=0 pw=0
time=12 us cost=2 size=559 card=43)(object id 40455)
         1          1          1               TABLE ACCESS BY INDEX ROWID MSC_PLANS (cr=2 pr=0 pw=0 time=11 us
cost=1 size=8 card=1)
         1          1          1                INDEX UNIQUE SCAN MSC_PLANS_U1 (cr=1 pr=0 pw=0 time=4 us cost=0
size=0 card=1)(object id 40420)
   1170347    1170347    1170347             PARTITION RANGE ITERATOR PARTITION: KEY (cr=5196 pr=0 pw=0
time=450106 us cost=2 size=0 card=1)
   1170347    1170347    1170347              INDEX RANGE SCAN MSC_SYSTEM_ITEMS_N3 PARTITION: KEY KEY (cr=5196
pr=0 pw=0 time=279759 us cost=2 size=0 card=1)(object id 149994)
  27537573   27537573   27537573         JOIN FILTER USE :BF0007 (cr=3 pr=0 pw=0 time=17282737 us cost=189707
size=462435600 card=4203960)
  27537573   27537573   27537573          PARTITION RANGE AND PARTITION: KEY(AP) KEY(AP) (cr=3 pr=0 pw=0
time=14430640 us cost=189707 size=462435600 card=4203960)
  27537573   27537573   27537573           TABLE ACCESS INMEMORY FULL MSC_BOM_COMPONENTS PARTITION: KEY(AP)
KEY(AP) (cr=3 pr=0 pw=0 time=1882537 us cost=189707 size=462435600 card=4203960)
  18112517   18112517   18112517      PARTITION RANGE ALL PARTITION: 1 25 (cr=146 pr=13 pw=0 time=3639446 us
cost=102248 size=435593444 card=16753594)
  18112517   18112517   18112517       TABLE ACCESS INMEMORY FULL MSC_ROUTING_OPERATIONS PARTITION: 1 25 (cr=146
pr=13 pw=0 time=2038064 us cost=102248 size=435593444 card=16753594)
```

cybernoor™

Demantra

- Improve performance of the Demantra key batch cycles as well as front-end worksheet performance
  - Forecast job
  - Workflow Background Engine
  - Front-end Worksheets (collaborator)
  - Safety Stock
  - Sales History

Demantra

- Improve performance of the Demantra key batch cycles as well as front-end worksheet performance

```
ALTER TABLE DP.SALES_DATA INMEMORY PRIORITY CRITICAL;

ALTER TABLE DP.MDP_MATRIX INMEMORY PRIORITY CRITICAL;
```

Demantra

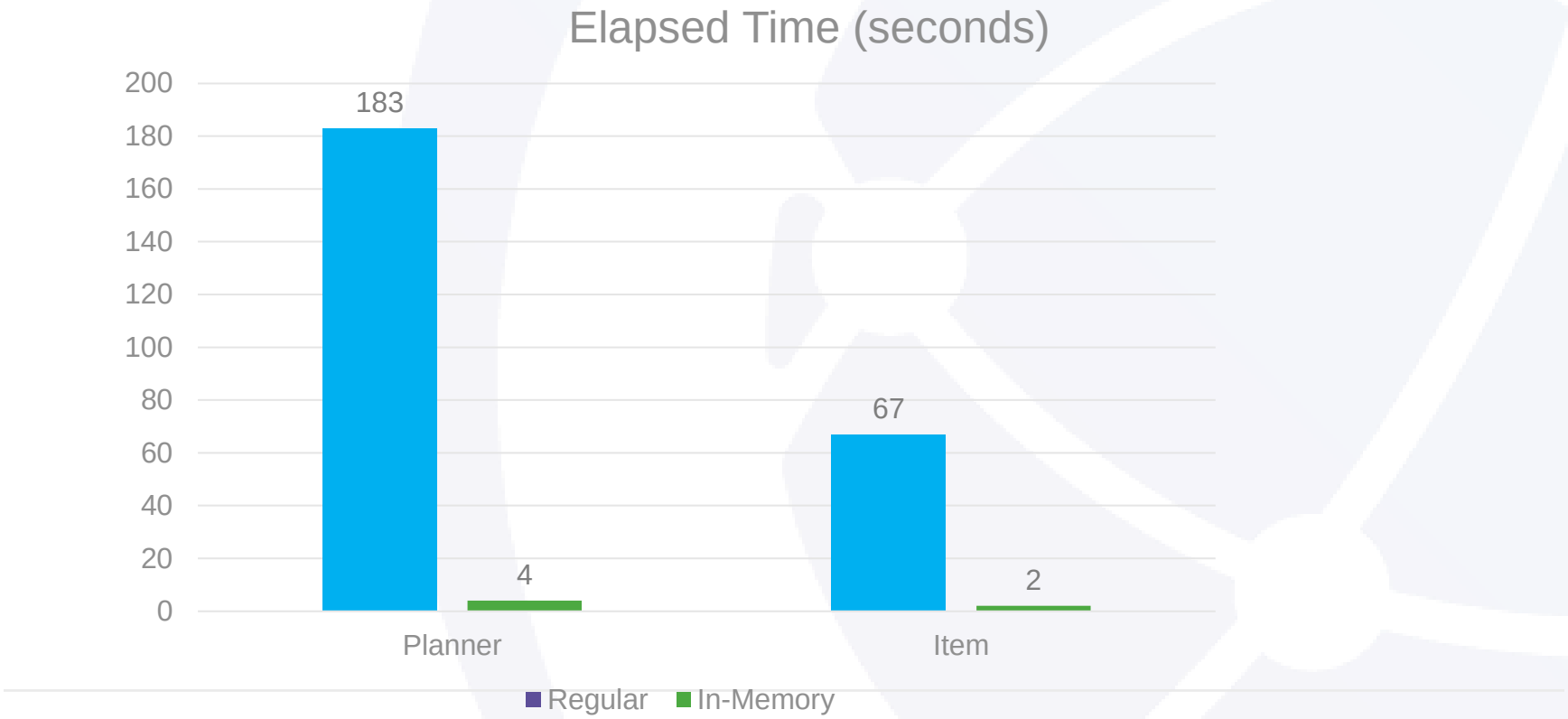
- Improve performance of the Demantra key batch cycles as well as front-end worksheet performance

```
SELECT MIN(NVL(ROWNUM,0)) AS COMB_ID, LEVEL1, LEVEL_ID1, LEVEL2, LEVEL_ID2, LEVEL3, LEVEL_ID3,
MIN(SEC_BASE0) AS SEC_BASE0, SUM(WEIGHT) AS WEIGHT, MAX(IS_MODIFIED0) AS IS_MODIFIED0
FROM SALES_DATA BRANCH_DATA
,MDP_MATRIX
,T_DATE_15249_38_1619915800 T_DATES_LIST
,T_EP_ITEM,T_EP_ORGANIZATION ,T_EP_EBS_FCST_GROUP
WHERE MDP_MATRIX.ITEM_ID = BRANCH_DATA.ITEM_ID
AND MDP_MATRIX.LOCATION_ID = BRANCH_DATA.LOCATION_ID
AND BRANCH_DATA.SALES_DATE >= T_DATES_LIST.FROM_SALES_DATE
AND BRANCH_DATA.SALES_DATE <= T_DATES_LIST.TO_SALES_DATE
AND BRANCH_DATA.ITEM_ID = MDP_MATRIX.ITEM_ID
AND BRANCH_DATA.LOCATION_ID = MDP_MATRIX.LOCATION_ID
AND MDP_MATRIX.T_EP_ITEM_EP_ID = T_EP_ITEM.T_EP_ITEM_EP_ID
AND MDP_MATRIX.T_EP_ORGANIZATION_EP_ID = T_EP_ORGANIZATION.T_EP_ORGANIZATION_EP_ID
AND MDP_MATRIX.T_EP_EBS_FCST_GROUP_EP_ID = T_EP_EBS_FCST_GROUP.T_EP_EBS_FCST_GROUP_EP_ID
AND (MDP_MATRIX.T_EP_SITE_EP_ID IN (2))
AND (MDP_MATRIX.T_EP_EBS_CPN_CODE_EP_ID IN (4))
AND (MDP_MATRIX.T_EP_I_ATT_4_EP_ID IN (50))
AND (MDP_MATRIX.T_EP_E1_IT_BR_CAT_3_EP_ID IN (222))
AND (MDP_MATRIX.FROM_DATE >= TO_DATE('03-30-2021 00:00:00','mm-dd-yyyy hh24:mi:ss')
OR MDP_MATRIX.UNTIL_DATE >= TO_DATE('03-30-2022 00:00:00','mm-dd-yyyy hh24:mi:ss'))
AND (MDP_MATRIX.FROM_DATE <= TO_DATE('04-04-2021 00:00:00','mm-dd-yyyy hh24:mi:ss')
OR MDP_MATRIX.UNTIL_DATE <= TO_DATE('04-04-2022 00:00:00','mm-dd-yyyy hh24:mi:ss'))
GROUP BY T_EP_ITEM.DM_ITEM_DESC
```

cybernoor™

Demantra

- Improve performance of the Demantra key batch cycles as well as front-end worksheet performance



cybernoor™

Summary

- Oracle Database In-Memory is a game changer – effective option to deal with the data growth challenges and deliver superior application performance.
- Review your application profile to determine which modules/objects make sense to leverage the IM store.
- Test thoroughly including batch cycles and volume data feeds to ensure performance SLAs are being met.
- Tune the in-memory configuration and review execution plans to ensure optimal plans are being used.

References

- Database In-Memory Guide (oracle.com)
- In-Memory Database | Oracle
- Oracle Blogs | Oracle Database In-Memory Blog
- MOS ID 2025309.1: Using Oracle Database In-Memory with Oracle E-Business Suite.
- MOS ID 2126233.1: Configuration and Troubleshooting Demantra and 12c In Memory Columns and Tables for Faster Read Transactions. Release 12.2.5.1 and Above.
- MOS ID 2252493.1: Understanding VCP Applications In-memory Planning on EXA Machine OR Using VCP 12.2.6.2 or Higher With 12c In Memory RDBMS

Q & A

Thank You