Oracle Database In-Memory How To Identify The Right Workload



May 15-16, 2019

Andy Rivenes Database In-Memory Product Management Oracle Corporation

Email: andy.rivenes@oracle.com

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





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 In-Memory: A Quick Recap



Row Format Databases vs. Column Format Databases



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



- 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

ORACLE

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

Oracle In-Memory Columnar Technology



- Pure in-memory column format
 - Enable for subset of database
 - Cheap to maintain no logging or IO
 - Allows efficient OLTP
 - No change to disk format
- Built **seamlessly** into Oracle Database
 - Appears as a new storage type
 - Transparent to Applications
 - All Enterprise Features work ..
 - Availability RAC, Flashback, DataGuard, etc.
 - Security Encryption, Auditing, etc.

In-Memory Columnar Technology



Optimizer Enhancements

Improves key aspects of analytic queries



- Speed of memory
- Scan and filter only the needed columns
- Vector Instructions



- Converts star joins into 10X faster column scans
- Scans large table for values that match the smaller table(s)

In-Memory Aggregation



- Creates an In-Memory aggregate accumulator
- Aggregates data during the fact table scan
- Runs Group By aggregations
 3-8x faster than non-IMA scans

Database In-Memory New Features



Performance

- In-Memory Expressions
- Join Groups
- In-Memory Dynamic Scans
- In-Memory Optimized Arithmetic



Managability

- Automatic Data Optimization
- Automatic In-Memory



Expanded Capacity

- Exadata Flash
- Active Data Guard
- External Tables

Why Use Database In-Memory





Improved Reporting Performance Faster Reports – No Application Changes

- Organizations can use Oracle reporting/analytical applications or existing 3rd party reporting tools
 - No application or data format changes required
- Improves performance (10x typical) of reporting applications with existing data warehouse and/or data marts
- Improves performance to ensure SLA's continue to be maintained
- Increases capacity of mixed workload environments to enable additional growth and performance





- Using Database In-Memory resulted in:
 - Triple the volume of Data
 - No changes required to Business Objects reports
 - 50X performance improvement on reports
 Benetic that took days new return in less than 1 hours
 - Reports that took days now return in less than 1 hour

Real-Time Analytics Use Operational Data for Real-Time Analytics

- Real-time analytics on operational data directly -without the time delay of moving data for reporting
- Enables real-time business intelligence at the point of contact
 - Delivers real-time insight, visibility and agility for critical business operations and decisions
- Enables real-time ad-hoc reporting /analysis and iterative drill-downs on operational data
- No application or data format changes required



Pricerite實惠

- Using Database In-Memory resulted in:
 - Analytic queries up to 5x faster
 - Real-time analytics dashboard

Reduced Overhead Faster Analytics -- Less Storage Overhead

- Analytic indexes can slow down the performance of transactional applications
 - Requires significantly more database storage (on costly tier 1 storage)
 - Increases overhead due to index maintenance
- Database In-Memory allows users to eliminate analytic reporting indexes – without impacting performance
- Removing the need for analytic reporting indexes greatly simplifies tuning and reduces ongoing administration



- Using Database In-Memory resulted in:
 - Performance Gains: 1.8X to 12X
 - Space savings and reduced contention on DML by dropping analytic indexes

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

How to Identify Analytic Workloads



What are Analytics?

an·a·lyt·i·cal

/ ana lidik(a)l/

adjective adjective: analytical

> relating to or using analysis or logical reasoning. "analytical methods" synonyms: systematic, logical, scientific, methodical, left-brained, (well) organized, ordered, orderly, meticulous, rigorous; diagnostic "the best chapters take a more analytical approach and try to work out some key principles" antonyms: unsystematic

Source: Google Search

• Our definition: Using aggregation to find patterns and trends in the data

ORACLE

What is an analytic query?



Only analytic queries benefit from accessing the IM column store

ORACLE

When and Where Should I Use Database In-Memory



ORACLE

Where to use In-Memory



- 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

- 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

ORACLE

Reporting

Database In-Memory - Use Cases







Mixed Workload

Real-Time Analytics

Reporting/BI

Which Objects Should I Populate





Oracle In-Memory Advisor

Workload Database Usage

ORACLE

Total Database Time	Analytics Processing Time	Analytics Processing
(Seconds)	(Seconds)	Percentage
2990	2640	88%

In-Memory Size	Percentage of Maximum SGA Size (100.0GB)	Estimated Analytics Processing Time Reduction (Seconds)	Estimated Analytics Processing Performance Improvement Factor
9.141GB	9%	2102	4.9X
8.684GB	9%	2101	4.9X
8.226GB	8%	2101	4.9X
7.769GB	8%	2100	4.9X

- In-Memory Advisor free download available on OTN for 11.2.0.3+ DBs
- 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 In-Memory Advisor

SQL Id	SQL Text	Analytics Processing Time Used (Seconds)	Estimated Analytics Processing Reduction (Seconds) With Unlimited Memory	Estimated Analytics Processing Performance Improvement Factor With Unlimited Memory	Estimated Analytics Processing Time Reduction (Seconds) With 9.1410B	Estimated Analytics Processing Performance Improvement Factor With 9.1410B
fp83uwmbzt8zd	select cf.custid, sum(act.purchase_amt) sales from all_card_trans act, cust_fact cf	990	696	3.4X	696	3.4X
7zkhj3xhq01w8	with gold_member_aff_cust as (select custid, aff_cc_num from cust_fact w	940	660	3.4X	660	3.4X
8p8ggufpp7699	with act as (select act.card_no, act.purchase_amt from all_card_trans act ,mcc m, zipcodes z	710	450	2.7X	450	2.78

Object Type	Object	Compression Type	Estimated In- Memory Size	Analytics Processing Seconds	Estimated Reduced Analytics Processing Seconds	Estimated Analytics Processing Performance Improvement Factor	Benefit / Cost Ratio (Reduced Analytics Processing / In-Memory Size)
TABLE	TEST_UNCOMP.ZIPCODES	Memory compress for query low	1.063MB	50	33	3.0X	507741 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_ME RCHANT_SALES.SYS_P5598.S YS_SUBP5592	Memory compress for query low	1.063MB	1	0	3.0X	36330 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_ME RCHANT_SALES.SYS_P5598.S YS_SUBP5593	Memory compress for query low	1.063MB	1	0	3.0X	36330 : 1
SUBPARTITION	TEST_UNCOMP.PARTNER_ME RCHANT_SALES.SYS_P5620.S YS_SUBP5615	Memory compress for query low	1.063MB	1	0	3.0X	28577 : 1

- Multiple sections available
 - In-Memory Size
 - SQL Statements with Analytic Benefit
 - Top object recommendations
 - All object based on memory size
 - Recommendation Rationale
 - Implementation SQL

ORACLE

Copyright $\ensuremath{\mathbb{C}}$ 2019, Oracle and/or its affiliates. All rights reserved. \mid

How Much Memory Do I Need





Oracle Compression Advisor And In-Memory

```
DECLARE
    l blkcnt cmp
                        PLS INTEGER;
    l blkcnt uncmp
                        PLS INTEGER;
                        PLS INTEGER:
      row cmp
                        PLS INTEGER;
    l row uncmp
                        PLS INTEGER;
    cmp ratio
    1 comptype str
                        VARCHAR2(100);
    comp ratio allrows NUMBER := -1;
BEGIN
    dbms compression. Get compression ratio (
    scratchtbsname => 'TS DATA',
                   => 'SSB',
    ownname
                   => 'LINEORDER',
    objname
    subobjname
                   => NULL
                    => dbms compression.comp inmemory query low
    comptype
    blkcnt cmp
                   => 1 blkcnt cmp
                      1 blkcnt uncmp,
    blkcnt uncmp
                   =>
    row cmp
                   =>
                      l row cmp,
    row uncmp
                   =>
                      1 row uncmp,
    cmp ratio
                   => cmp ratio,
    comptype str
                   => 1 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
- Uses DBMS_COMPRESSION
- Applies MEMCOMPRESS to sample set of data from a table
- Returns estimated compression ratio
- Requires 12.1.0.2 or higher

What Queries Benefit From Database In-Memory?





Areas that Benefit from Database In-Memory

 Data Access for Analytics and Reporting

 Analytic Index Maintenance

ORACLE

More Information:
 <u>When to Use Oracle Database In-</u>
 <u>Memory</u>



Abstract Time Profile for a Typical Application (* - time potentially reduced by Database In-Memory)

Areas that DON'T Benefit from Database In-Memory

- Application Time
- Network communication
- Logon and Logoff
- Parse Time
- PL/SQL functions
- OLTP Queries
- OLTP index maintenance
- Writes to Database Tables



What SQL Techniques Can Increase The Benefit of Database In-Memory?



ORACLE

SQL Techniques That Can Improve Benefit

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



What SQL Techniques Can Reduce the Benefit of Database In-Memory?



ORACLE

Hints can prevent the Optimizer from Choosing In-Memory

- Hints that force the Optimizer to choose a join method
- Hints that force the Optimizer to use an Index
- Hints that result in the creation of temporary tables (do not benefit from Database In-Memory)
- Hints that disable In-Memory access
- Hints that disable features (i.e. NO_PARALLEL, NO_REWRITE)



SQL Techniques That Can Reduce Benefit

- Sub-query factoring (WITH) Can prevent optimial scan filtering typically used as a generic row source (MVs with query rewrite may be a better choice)
- Use of Function-based Indexes prevents predicate push down, has to be evaluated after the scan
- Common views often accesses more data than needed for the query
- Nested Views can be difficult for the Optimizer to efficiently unnest
- Correlation issues unless using extended statistics, the Optimizer may calculate the wrong cardinality

What Query Characteristics Benefit From Database In-Memory?



ORACLE

Which Queries Benefit From Database In-Memory?

For a non-trivial amount of rows and execution time, when a significant amount of time ...



ORACLE

Use Time Based Analysis Techniques To Evaluate Benefit SQL Monitor Active Reports

- Shows how SQL was executed and where time was spent
- See blogs.oracle.com/In-Memory for a technical brief on creating SQL Monitor active reports

leneral						Time & Wait St.	atistics			30.9	atistics			
becution Started	558	P14	re(*), 5			Duration Database Time PL/SQL 8. Jave Activity %	0s		1.5x 0.5x 900	10 R	er Getta Ispuestta D Bytes			,
Details Plan Batistics	Rome In At	uty												
Pan Batistics	2067213421		u.,	bine-	Cast	Timeline(1x)	feet.	Achuel	Pana.	Serie (6. K) Re	10	Advity %
Plan Batlatics Plan Hash Volue	226/72129621	Note	u	bine.	Cost	Tonative(1a)	6m	Actual	Rena.	Ί ετφ (6. X) No	10	Advely %
Plan Ballation Plan Hash Value Operation	2047213421 🔛 Par	Note	U 0 1	Estimate	Gast	Tinaline(1a)	1m.	Adual	Harrar .	Samp (0. H) Re	10	Aduly %
Plan Batistics Plan Hash Value Operation © SELECT STA © SOLT AGO	2047213421 🔛 Par	Note	0. 0 1 2	Estimute		Totaline(1x)	1 1 1	Actual	Here.	Yenge (6. X) Ba	8-	Autority %

ORACLE
Data Scans





Scanning & Filtering Query

• Query to list total number of orders and the total value of merchandise shipped by air.

SELECT

```
count(*),
SUM(l.lo_ordtotalprice)
FROM lineorder l
WHERE l.lo shipmode = 'AIR';
```



Scanning & Filtering Query: Traditional data access

eneral						Time & Wait Stal	tistics			10 58	atistics				
Secution Started	SELECT /*+ NO_PARALLEL M Thu Jan 12, 2017 1:54:16 PM Thu Jan 12, 2017 1:54:21 PM 16777217	1	t(*), S	U		Duration Duration Duration Duration Duration PL/SQL & Java 0 Activity %	s		5.0s 4.7s	10 Re	or Gets Quests Bytes			187K	
User Fetch Calls													-	ority of nt acces	
Details														data	
Plan Statistics Plan Hash Value	1385010855 📄 Plan No		U	Estimate	Cost	Timeline(5s)	Exec	Actual	Memor	Temp (0 10	Re)	0 Adt		
Plan Statistics Plan Hash Value	1385010855 🧧 Plan No	ote		Estimate	Cost	Timeline(5s)	Exec	Actual	Memor	Temp (0 10	Re)	0 Add		
Plan Statistics Plan Hash Value Operation	1385010855 Plan No	ote	U	Estimate	Cost	Timeline(5s)	Exec	Actual 1	Memor	Temp (0 10	Re 3	0 Add		
Plan Statistics Plan Hash Value Operation - SELECT STAT - SORT AGG	1385010855 Plan No	Name	U 0	Estimate 1 10M	Cost 218K	Timeline(5s)	Exec 1 1	1	Memor	Temp (0 10	Re 1	0 Act	- 75	

ORACLE

Scanning & Filtering Query: Scan & filter data in-memory

eneral						Time & Wait Stat	istics			10 58	atisti	ics		
Execution Started	SELECT /*+ NO_PARALLEL Thu Jan 12, 2017 1:51:44 / Thu Jan 12, 2017 1:51:45 / 16777216	PM	nt(*), S	U		Duration Database Time PL/SQL & Java 0: Activity %	;		1.0s 0.9s 100	IO Re	er Gets quests) Byte	s 0		3
User Fetch Calls Details	1												spe	ime now nt scanni Itering da
Plan Statistic	s 🔆 Plan 📐 Activ 2267213921 📄 Plan			Fatimata	Cost	Timeline(1s)	Exec	Actual	Memor	Temp (0	IO Re		
		Name	U	Estimate										y %
	TEMENT	Name	0	Esomate			1	1						19 m
Operation		Name	_	Esomate			1	1						4.2

ORACLE

Things To Remember With Scans

- Queries that spend a significant amount of time accessing data are good candidates for Database In-Memory
- What Queries benefit most?
 - Queries that select just a subset of the columns from a table
 - Queries with selective where clause filter predicates
 - Queries with equality predicates, in-lists or range predicates work best
 - Queries with like or not exists predicates don't see as much benefit
- 12.2 Adds
 - In-Memory Expressions Avoids repeated expression evaluation
 - More predicate push down operations improves scan performance



Joins



Join Query

• Query to show total revenue by brand

```
SELECT p.p_brand1,
	SUM(lo_revenue) rev
	FROM lineorder l,
	part p,
	supplier s
	WHERE l.lo_partkey = p.p_partkey
	AND l.lo_suppkey = s.s_suppkey
	AND p.p_category = 'MFGR#12'
	AND s.s_region = 'AMERICA'
	GROUP BY p.p_brand1
```

Join Query – Traditional hash join

leneral						Time & Wait Sta	distics				10 Stat	istics				
Execution ast Refre Exec	n Started	558	NO_PARALLEL	[Duration Database Time PL/SQL & Jave Activity %	8			8.0s 7.8s	Buffer (10 Requ 10 B				33	
Details															ajority ent joir	
Plan	Statistics	Activity 🔀 Metrics												\sub		0
Plan Ha		1153046401	Name	Étő	Cost	Timeline(8s)	Exec	Actual Rows	Memory	Temp (Other	10 Req	10 Bytes	5		
Plan Ha	operation	1153046401	Name	Бб	Cost	Timeline(8s)	Exec	Actual Rows 40	Memory	Temp (Other	10 Req	10 Bytes	5		
Plan Ha Line	Operation	1153046401	Name	Est 1,000	Cost 129K	Tindina(8s)	Eec 1		Memory 1MB	Temp (Other	10 Req	10 Bytes	5		
Plan Ha Line	Operation	1153046401 ECT STATEMENT	Name			Timeline(8s)	Esec 1 1	40		Temp (Other	10 Req	10 Bytes	Activity %		
Plan Ha Line	Operation - SELI - HP	1153046401 ECT STATEMENT ASH GROUP BY HASH JODN TABLE ACCESS INMEMORY FULL	Name	1,000	128K	Timeline(Bs)	Exec 1 1 1	40	148	Temp (10 Reg	30 Bytes	Activity %		
Plan Ha Line	Operation - SELI - HP	1153046401 ECT STATEMENT ASH GROUP BY HASH JOIN TABLE ACCESS INMEMORY FULL 3 HASH JOIN	SUPPLIER	1,000 842K 380K 4,192K	128K 128K 237 117K	Timeline(8s)	Exec 1 1 1 1 1	40 40 584K 380K 2,922K	148	Temp (10 Reg	30 Bytes	Activity %		0
Plan Ha Line	Operation - SELI - HP	IIIS3046401 ECT STATEMENT ASH GROUP BY HASH JOON TABLE ACCESS INMEMORY FULL HASH JOON	SUPPLIER PART	1,000 842K 380K 4,152K 80K	128K 128K 237 117K 212	Timeline(8s)	Exec 1 1 1 1 1 1 1	40 40 584K 380K 2,922K 80K	1MB 16MB	Temp (10 Req	30 Bytes	Activity %		
Plan Ha Line	Operation - SELI - HP	1153046401 ECT STATEMENT ASH GROUP BY HASH JOIN TABLE ACCESS INMEMORY FULL 3 HASH JOIN	SUPPLIER PART	1,000 842K 380K 4,192K	128K 128K 237 117K	Timeline(8x)	Exec 1 1 1 1 1 1 1 1	40 40 584K 380K 2,922K	1MB 16MB	Temp (10 Req	10 Bytes	Activity %		0

ORACLE

Join Query – In-Memory hash join with Bloom filters

eneral	wi				Time & Wait Statistic	3			10 Statisti	C8			
	SQL Text. SELECT /*+ NO_PARALLEL NO_VECT	OR_TRANSFO	-		Duration	_	_	4.0s	Buffer Geb				
secutio	ion Started Thu Jan 19, 2017 4:09:10 PM		_	1	Database Time			3.75	30 Requests	0			
ast Ref	fresh Time Thu Jan 19, 2017 4:09:14 PM				PL/SQL & Jeve Ds				30 Byte	s 0			
De	secution ID 16777216				Activity %			100					
	User 558 Fetch Calls 4								Лаjority	of tir	ne		
	Peter Carls												
									now s	spent			
Details	la la								scann	ing &			
Plan	n Statistics 🔆 Plan 📐 Activity [Metrics								-			
-		Hetrics							filterin	-)	
Plan H	Hash Value 322496559 Plan Note								filterin	g dat	a 🦯)	
Plan H	Hash Value 322496559 Plan Note Operation	Name	Estin	Cost	Timeline(4s)	Exec		Memory (Max)		g dat	a 🦯	Activity %	
Plan H	Aush Value 322496559 Plan Note Operation E) SELECT STATEMENT				Timeline(4s)	Exec	40		filterin	g dat	a 🦯	Activity %	
Plan H	Aush Value 322496559 Plan Note Operation El SELECT STATEMENT El HASH GROUP BY		1,000	128K	Timeline(4s)	Exec	40 40	148	filterin	g dat	a 🦯		
Plan H	Hash Value 322496559 Plan Note Operation SELECT STATEMENT HASH GROUP BY HASH JOIN			128K	Timeline(4s)	Exec 1 1	40		filterin	g dat	a 🦯	Adivity %	
Plan H	Aush Value 322496559 Plan Note Operation El SELECT STATEMENT El HASH GROUP BY		1,000	128K	Timeline(4s)	Exec 1 1 1 1	40 40	148	filterin	g dat	a 🦯		
Plan H	Hash Value 322496559 Plan Note Operation SELECT STATEMENT HASH GROUP BY HASH JOIN	Name	1,000 842K	128K	Timeline(4s)	Exec 1 1 1 1 1	40 40 584K	148	filterin	g dat	a 🦯		
Plan H ine 0 1 2 3	Auch Value 322496559 Plan Note Operation SELECT STATEMENT HASH GROUP BY HASH JOIN TABLE ACCESS INMEMORY FULL	Name	1,000 842K 380K	128K 128K 237	Timeline(4s)	Esec 1 1 1 1 1 1	40 40 584K 380K	146 1646	filterin	g dat	a 🦯		
Plan H ine 0 1 2 3 4	Hash Value 322496559 Plan Note Operation SELECT STATEMENT HASH GROUP BY HASH JOIN TABLE ACCESS INMEMORY FULL HASH JOIN	SUPPLIER IBF0000	1,000 842K 380K 4,192K	128K 128K 237 117K	-	Exec 1 1 1 1 1 1 1 1 1	40 40 584K 380K 2,922K	146 1646	filterin	g dat	a 🦯		
Plan H ine 0 1 2 3 4 5	Hash Value 322496559 Plan Note Operation SELECT STATEMENT D HASH GROUP BY D HASH JOIN TABLE ACCESS INMEMORY FULL D HASH JOIN D HASH JOIN D JOIN FILTER CREATE	SUPPLIER IBF0000	1,000 842K 380K 4,152K 80K	128K 128K 237 117K 212	-	Exec 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	40 40 584K 380K 2,922K 80K	146 1646	filterin	g dat	a 🦯		

ORACLE

Things To Remember With Joins

- Queries that spend a significant amount of time joining data are good candidates for Database In-Memory
- Work best with selective equality joins
 - Generates Bloom filter(s)
- Non equality joins won't benefit as much
 - Consider using materialized views
 - Make sure you have enough PGA allocated
 - Consider using parallel execution
 - 12.2 Adds Join Groups can improve join performance by using common dictionary encodings

Aggregation





Aggregation Query

• Query to show total profit by year and nation

<pre>SELECT d.d_year, c.c_nation, sum(lo_revenue - lo_supplycost)</pre>
FROM LINEORDER 1, DATE_DIM d, PART p, SUPPLIER s, CUSTOMER c
WHERE l.lo_orderdate = d.d_datekey
AND l.lo_partkey = p.p_partkey
AND l.lo_suppkey = s.s_suppkey
AND l.lo_custkey = c.c_custkey
AND s.s_region = 'AMERICA'
AND c.c_region = 'AMERICA'
GROUP BY d.d_year, c.c_nation
ORDER BY d.d_year, c.c_nation;

ORACLE

Aggregation – Traditional Group By

neral		1	me & Walt Stati	atica				10 Start	intica						
SQL Test: MELECT /*+ NO_HWALLEL NO_VECTOR_ seculium Started: Web Jan 36, 2017 9:13-22 PF at Refresh Time: Web Jan 18, 2017 9:29 20 PF	TRANSPO	0	Duration III	_			15.0m 25.0m	Buffer (10 Repu 30 B	_				100		
Descular 10 18777218 Unor 1958 Petch Calls 4			Addivity %				100				C				cked up by
befails													oper	atio	n
Plan Statiatica 🥀 Plan 📐 Athiliy 🛄 He	eics.											_			
Pan hash Value 16311963 - Pian Note															
Operation	Name	Line	Estimated R	Cast	Timeline(H01a)	Decul-	Adual Rova	Henory (Temp (Max)	0 30 Res	uesta 10 B	- ADV			
E SELECT STATEMENT							35				-			_	
E SORT GROUP BY		1	124	184			35	48			1	-	8		
E HASH JOIN		2	3394	184			228H	248				11			
THRUE ACCESS FULL	DATE_DOM_NUMBER	3	2,556	7		1	2,596								
ED HARM XOON			33994	184			226H	9840					2		
TABLE ACCESS INMEMORY PULL	PART	5	2,000K	187		1	2,000K								
@ 1969 X00V		- 6	339M	1790			228M	37448					12		
TABLE ACCESS INHERORY FULL	CLETOHER.	7	1,7128	3,434		1	5,702K								
(5) HAGH JOON			1.116M	13M			1,1394	1748					18		
- JOIN FILTER CREATE	:8P0000		380K	2,471		1	380K								
A AGENE	VW_G8F_28	- 18		2,471		1	360K						_		
HASH GROUP BY		11		2,471		1	3804	1448							
	SUPPLIER	- 12	3804	237		1	3004								
THRUE ACCESS INMEMORY PULL		13	5,700M	\$57K			1,5344								
THRUE ACCESS INMEMORY FULL D-JOIN FILTER USE	:8F0000		Street,												

ORACLE

Aggregation – Vector Group By with In-Memory

Seneral	Time & Walt	Relation					10 Subility					
50, Text: SELECT /*= N0, MARLEL HONTOR */ 6.6, year or, bander Darted Web Jer 18, 2017 10:30:04 /# Bander Tres: Web Jer 38, 2017 10:40:00 /# Bander 10: 10:77314 Bander 10: 10:77314 Bander 2018	Contraction for the second sec					5.3m 5.3m	Mitr Gen II 10 Repuesto di 10 Rysee di				70	
Details												2
Par Relation An Par L Adulty P. Maria												
Par Net York 124840400												
Line ID Operation	Same	Estimated Rove	Cast	Train(20h)	Deculiars	Aduat Rove	Henory (Mar)	Temp (Marc)	OL. 10 Repuesto	10 Bytes	Activity %	
B BELET EXTENSION					1	35						
1 B TEMP TABLE TRANSFORMATION						35						
2 B LOAD AS SELECT (DURSON DUARTION HERORY)	\$15,709,000084				1	1	210					
8 B IDAD AS BELICT (DURSON DUARTION HERORY)	DIS_TONP_DYDROMBA				1	1	10			-		
10 B LOAD AS SELECT (CLASOR DUARTION HERORY)	\$15,704P,(F040484						10				С	
14 (B LOAD AS SELECT (DURSOR DURATION HEMORY)	\$15,7049,\$7040484				1	1	28				Jroup By	is now a sca
18 gi solt diour ar			2274		1	28	40					
10 gb Herder JOON			1274			28	6090				& me	r operation
20 TABLE ACCESS P.A.L	\$15,704P,\$P040484	8	2		1							
21 @ HADH JOIN			3574			18	118			-		
22 @ HERGE JOIN CARTEENIN					1							
28 <u>()</u> VEW	V8.37,89725627		528		1	38						
30 IB VECTOR GROUP BY			9274			18	24/8					
31 (B) HADH GROUP BY			10.74									
32 @ HEY VECTOR USE	404000	2004	12.74			2784					1.30	
30 gb HEY HOCTOR USE	10/5001	2004	528			2284						
24 (g) KEY VECTOR LISE	10/1003	100	5.76		-	1284					V	
IN ID NOT VECTOR USE	10/0002	1,1174	10%			2284					V	
36 TABLE ACCESS INVENDED FALL	UNIDEDER, SINGLE	1,7004	10.74			1000						

ORACLE

Things To Remember With Aggregation

- Queries that spend a significant amount of time aggregating data are good candidates for Database In-Memory
- Vector Group By works best when
 - INMEMORY_SIZE parameter must be set to a non-zero value
 - The join columns between the tables contain "mostly" unique keys or numeric keys
 - The fact table (largest table in the query) is at least 10X larger than the other tables
 - The tables are populated into the IM column store
- NOTE: Tables don't have to be in the IM column store vector group by can also be offloaded to Exadata storage



How Have Customers Benefited From In-Memory





How Customers Use Database In-Memory

AT&T WiFi – Data Warehouse Seated Wi-Fi	Villeroy & Boch – SAP BW
 Business Objects reports 100X faster 	 SAP BW COPA queries 30 – 33X faster
 ETL processes improved by 50% faster 	 SAP Transaction list queries 4 – 4,800X faster
 No changes to SAP Business Objects reports 	 Avoided expensive & risky upgrade to S4/Hana
 BOSCH – SAP CRM (B) BOSCH Dropped all custom indexes Analytic queries 2-20X faster, DML 2-3X faster No changes to application required 	 Die Mobiliar – Mixed Workload <i>Die</i> Mobiliar Analytic queries 50-200X faster Database size reduced considerably Phase out of Netezza and mainframe systems

How Customers Use Database In-Memory

 Mankind Pharma – Mixed Workload Analytical reports 11x faster Dropping indexes improved OLTP 90% reduction in database size 	 Shanghai Customs – Mixed Workload Processes Clearance 43x Faster Improves Declaration-Services Efficiency Reduced Costs
 LION – SAP ERP Analytic queries 4X faster Transactions 2X faster Analytic queries now possible on 100s of Millions of Point-of-Sales Transactions 	 Lufthansa – Reporting Application Content Sectors Analytic queries up to 100x faster Improved data ingest performance Reduction in database size

Database In-Memory - Use Case Summary



How Do I Get Started



ORACLE

https://blogs.oracle.com/in-memory/dbim-resources



ORACLE

Additional Resources



Join the Conversation

- https://twitter.com/db_inmemory
- https://twitter.com/TheInMemoryGuy
- http://www.oracle.com/goto/dbim.html

Database In-Memory Information

Database In-Memory Blog

oracle.com – Database In-Memory

Database In-Memory YouTube Channel

Ask TOM Database In-Memory Office Hours

Database In-Memory Guide (Documentation)

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |