ORACLE



Oracle Database In-Memory

Best Practices for Getting Started, HrOUG 2021



Andy Rivenes

Database In-Memory Product Management Oracle Corporation Email: <u>andy.rivenes@oracle.com</u> Twitter: @TheInMemoryGuy Blog: <u>blogs.oracle.com/in-memory</u>

Just The Highlights

- This presentation will just briefly cover the highlights
- More details available on the Database In-Memory Resources page:
 - https://blogs.oracle.com/in-memory/dbim-resources
 - Database In-Memory Quick Start Guide
 - Database In-Memory Implementation Guidelines
- Don't forget the documentation:
 - Database In-Memory Guide

Organizations have many business questions

Yet critical reporting and analytics don't perform at business speeds



Introducing Database In-Memory

What's your favorite data format?



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

Where Is Database In-Memory 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:
 - Database Cloud Service Virtual Machines: Extreme Performance
 - Database Cloud Service Bare Metal: Extreme Performance
 - Exadata Cloud Service
 - Exadata Cloud at Customer
 - Autonomous Data Warehouse (Flash only)
 - On-premises
 - Oracle Database XE



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

How Do You Know If You Will Benefit From Database In-Memory?

Oracle In-Memory Advisor



Workload Database Usage

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 <u>oracle.com</u> 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 M	SQL Text	Analytics Processing Time Used (Seconds)	Estimated Analytics Processing Time Reduction (Seconds) With Unifiented Memory	Estimated Analytics Processing Performance Improvement Factor With Unlimited Memory	Estimated Analytics Processing Time Reduction (Seconds) With 0.14108	Estimated Analytics Processing Performance Improvement Factor With S.14308	
fp83uwmbet8ed	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 mee m, zipcodes z	710	450	2.7X	450	2.7X	

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

Database / Technologies / Oracle Database In-Memory Advisor Dracle Database In-Memory Advisor Oracle Database In-Memory Advisor Oracle Database In-Memory allowing a single database to efficiently support

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

Use A Current Version of Oracle Database

Installing Oracle Database In-Memory

	Oracle Database 19c Installer - Step 9 of 10	_ = ×
nstall Product		19° ORACLE Database
Configuration Option Database Installation Options Database Edition Installation Location	Progress 11% Linking RMAN Utility	
Operating System Groups Root script execution Prerequisite Checks Summary Install Product Finish	Status Configure Local Node Configure Local Node Prepare Link binaries Setup Setup Setup Oracle Base Execute Root Scripts	In Progress Succeeded In Progress Pending Pending Pending
	Details Revert A	II R <u>e</u> vert <u>R</u> etry <u>S</u> kip
Help	19° ORACLE' Database	

- Automatically installed as part of Oracle Database
- Strongly recommend using a current version (19c or 21c)
- No additional steps required

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

Installing: Apply the Latest Database Proactive Bundle Patch or Release Update

- Database In-Memory fixes and enhancements are only distributed through Database Proactive Bundle Patches or Release Updates
- See MOS Notes:
 - 2337415.1 Overview of Database Patch Delivery Methods for 12.2.0.1 and greater
 - 1962125.1 Overview of Database Patch Delivery Methods for 12.1.0.2 and older
- Starting with the latest patches avoids re-inventing the wheel discovering bugs that have already been fixed!



What About Upgrades?

Database Parameter Settings

- Evaluate all non-default initialization parameters
- Strongly consider unsetting any underscore parameters unless you are sure they are needed



Don't Steal Memory!

In-Memory Area: Static Area within SGA



- Contains data in the new In-Memory Columnar Format
- Controlled by INMEMORY_SIZE
 parameter
 - Minimum size of 100MB
- Can be re-sized larger while database is running (starting in12.2)
- SGA_TARGET must be large enough to accommodate In-Memory area



Note: Don't steal Memory from other components

Configuring : In-Memory Column Store

Don't get carried away

SOL ID 7isk74a56hm2x (I)	Time	& Wait S	tatist	ics				10	Statistics			
Execution Started Tue Mar 6, 2012 10	57:26 AM	Duratio	n 📕				10.5m		Buffer Gets			1,352K
Last Refresh Time Tue Mar 6, 2012 11: Execution ID 16777217	07:58 AM Dal	tabase Tim	e 📕				10.5m		IO Requests	86	ж	
User TANEL	PL,	/SQL & Jav	a 0.0	Os					10 Bytes	_	_	26GB
Fetch Calls 0	Wai	t Activity 9	6				100	0	ell Offload Efficiency	-14.9	94%	
Details												
	-											
Plan Statistics 🖓 Plan 📐 Activi	ty Retrics											
	. 100											
Ian Hash Value 147847724	. [00					0	TIP: Rig	ht mouse	e click on the table allo	wws to to	ggle between 10 R	equests and 10 Byt
Ian Hash Value 147847724 Operation	Name	Estim	Cost	Timeline(63	Ex	Act	Me	ht mouse Tem	e click on the table allo 10 Bytes	Cell	CPU Activity	Wait Activity.
Ian Hash Value 147847724 Apperation CREATE TABLE STATEMENT	Name	Estim	Cost	Timeline(63	Ex	@ Act	Me	ht mouse Tem	e click on the table allo IO Bytes	Cell	ggle between 10 Rd CPU Activity	Quests and 10 Byt
I AT Hash Value 147847724 speration CREATE TABLE STATEMENT D LOAD AS SELECT	Name	Estim	Cost	Timeline(63	Ex 1 1	@ Act 1	Me	ht mouse	e dick on the table allo IO Bytes 985MB	Cell	CPU Activity	Wait Activity.
CREATE TABLE STATEMENT	Name	Estim	Cost 2,45	Timeline(63	Ex 1 1 1	Act 1 1 26M	529KB 816ME	Tem 2GB	e dick on the table allo IO Bytes 985MB 4GB	Cell	CPU Activity 202 46	2.7
CREATE TABLE STATEMENT LOAD AS SELECT HASH GROUP BY HASH JOIN	Name	Estim 135M 135M	Cost 2,45 7271	Timeline(63	Ex 1 1 1 1	Act 1 1 26M 135M	529KB 816ME 648ME	Tem 2GB 5GB	e dick on the table allo IO Bytes 985MB 4GB 9GB	Cell	ggle between 10 R/ CPU Activity 1 02 46 32	Wait Activity.
Ian Hash Value 147847724 Apperation CREATE TABLE STATEMENT LOAD AS SELECT HASH GROUP BY HASH JOIN HASH JOIN	Name	Estim 135M 135M 45M	2,45 7271 3251	Timeline(63	Ex 1 1 1 1 1	Act 1 1 26M 135M 45M	TIP: Rig Me 529K8 816ME 648ME 415ME	Tem 2GB 5GB 2GB	e dick on the table allo IO Bytes 985MB 4GB 9GB 3GB	Cell	CPU Activity 46 32 17	Wait Activity.
an Hash Value 147847724 Apperation CREATE TABLE STATEMENT LOAD AS SELECT HASH GROUP BY HASH JOIN HASH JOIN TABLE ACCESS STORAG.	Name	Estim 135M 135M 45M 45M	2,45 7271 3251 96K	Timeline(63	Ex 1 1 1 1 1 1	Act 1 26M 135M 45M 45M	529KB 816ME 415ME	Tem 2GB 5GB 2GB	e dick on the table allo IO Bytes 965MB 4GB 9GB 3GB 3GB	Cell	CPU Activity 46 32 17 67	2.7 19 26
An Hash Value 147847724 Apperation CREATE TABLE STATEMENT LOAD AS SELECT HASH GROUP BY HASH JOIN HASH JOIN TABLE ACCESS STORAG. TABLE ACCESS STORAG.	Name Name ORDERS CUSTOMERS	Estim 135M 135M 45M 45M 45M	2,45 7271 3251 96K 1051	Timeline(63	Ex 1 1 1 1 1 1 1 1 1	Act 1 1 26M 135M 45M 45M 45M 40M	TIP: Rig Me 529KB 816ME 648ME 415ME	Tem 2GB 5GB 2GB	e dick on the table allo IO Bytes 985M8 4GB 9GB 3GB 3GB 3GB 3GB	Cell 72 59	202 46 32 17 67 .34	Wait Activity.
In Hash Value 147847724	Name Name ORDERS CUSTOMERS ORDER_ITEMS	Estim 135M 135M 45M 45M 45M 135M	2,45 7271 3251 96K 1051 1341	Timeline(63	Ex 1 1 1 1 1 1 1 1 1 1	Act 1 1 26M 135M 45M 45M 45M 40M 135M	529KB 816ME 415ME	tt mouse Tem 2GB 2GB 2GB	e dick on the table allo IO Bytes 985M8 4G8 9GB 3G8 3G8 3G8 4G8	72 59 38	202 46 32 17 57 .34 1.01	Wait Activity.

- Don't give all memory to SGA
- Don't want any aspect of execution plan to spill to disk
- Ensure PGA_TARGET is set large enough to keep joins & sorts in memory
- Use Parallel Execution to maximize PGA usage

Tip: Column Store Sizing

Allocate Extra Room



- The column store is fixed in size and objects are fully populated, but ...
 - Inserts will cause the object to grow when the new rows are populated
 - Updates can cause existing rows to expand (i.e. column values) which can affect the size of a repopulated IMCU
- Additional space should be reserved in the column store to allow for DML activity

Fully Populate Your Data

Tip: View In-Memory Area Usage

V\$INMEMORY_AREA: Current size of pools in the In-Memory area

V\$IM_SEGMENTS: List of segments currently populated in the In-Memory column store SQL> SELECT * FROM v\$inmemory area;

POOL	ALLOC_BYTES	USED_BYTES	POPULATE_STATUS
1MB POOL	5,179,965,440	3,241,148,416	DONE
64KB POOL	570,425,344	9,568,256	DONE

SQL> SELECT owner, segment_name, populate_status, inmemory_size, bytes_not_populated FROM v\$im segments;

OWNER	NAME	STATUS	In-Memory Size	Populated
SSB	LINEORDER	COMPLETED	3,206,086,656	0
SSB	DATE DIM	COMPLETED	1,179,648	0
SSB	SUPPLIER	COMPLETED	2,228,224	0
SSB	PART	COMPLETED	18,022,400	0
SSB	CUSTOMER	COMPLETED	23,199,744	0

Monitoring : In-Memory Column Store Population V\$IM_SEGMENTS

- Indicates which objects are currently populated in-memory
- Shows current size of each segment in-memory
- Shows how much remains to be populated

SQL> select segment_name, populate_status, inmemory_priority, inmemory_size, bytes_not_populated from v\$im_segments;

SEGMENT_NAME	POPULATE_STATUS	INMEM_PRIORITY	INMEM_SIZE	BYTES_NOT_POPULATED
ACCOUNTS	STARTED	HIGH	196606	2434886912
SALES	COMPLETED	CRITICAL	135790592	0

What If You Don't Have Enough Memory?

Oracle Compression Advisor And In-Memory

DECLARE

v_blkcnt_cmp	BINARY_INTEGER;
v blkcnt uncmp	BINARY INTEGER;
v row cmp	BINARY_INTEGER;
v row uncmp	BINARY_INTEGER;
v_cmp_ratio	NUMBER := -1;
v_comptype_str	VARCHAR2(60);
BEGIN	
DBMS_COMPRESSION.	.GET_COMPRESSION_RATIO (
scratchtbsname	=> 'TS_DATA',
ownname	=> 'SSB',
objname	=> 'LINEORDER',
subobjname	=> NULL,
comptype	=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW,
comptype blkcnt_cmp	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkcnt_cmp,</pre>
comptype blkcnt_cmp blkcnt_uncmp	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkcnt_cmp, => v_blkcnt_uncmp,</pre>
comptype blkcnt_cmp blkcnt_uncmp row_cmp	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkent_emp, => v_blkent_unemp, => v_row_emp,</pre>
comptype blkcnt_cmp blkcnt_uncmp row_cmp row_uncmp	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkcnt_cmp, => v_blkcnt_uncmp, => v_row_cmp, => v_row_uncmp,</pre>
comptype blkcnt_cmp blkcnt_uncmp row_cmp row_uncmp cmp_ratio	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkent_emp, => v_blkent_unemp, => v_row_emp, => v_row_unemp, => v_emp_ratio,</pre>
comptype blkcnt_cmp blkcnt_uncmp row_cmp row_uncmp cmp_ratio comptype_str	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkcnt_cmp, => v_blkcnt_uncmp, => v_row_cmp, => v_row_uncmp, => v_cmp_ratio, => v_comptype_str,</pre>
comptype blkcnt_cmp blkcnt_uncmp row_cmp row_uncmp cmp_ratio comptype_str subset_numrows	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkcnt_cmp, => v_blkcnt_uncmp, => v_row_cmp, => v_row_uncmp, => v_cmp_ratio, => v_comptype_str, => DBMS_COMPRESSION.COMP_RATIO_ALLROWS);</pre>
comptype blkcnt_cmp blkcnt_uncmp row_cmp row_uncmp cmp_ratio comptype_str subset_numrows DBMS_OUTPUT.PUT_I	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkcnt_cmp, => v_blkcnt_uncmp, => v_row_cmp, => v_row_uncmp, => v_cmp_ratio, => v_comptype_str, => DBMS_COMPRESSION.COMP_RATIO_ALLROWS); JINE('Compression Type: ' TO_CHAR(v_comptype_str));</pre>
comptype blkcnt_cmp blkcnt_uncmp row_cmp row_uncmp cmp_ratio comptype_str subset_numrows DBMS_OUTPUT.PUT_I DBMS_OUTPUT.PUT_I	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkcnt_cmp, => v_blkcnt_uncmp, => v_row_cmp, => v_row_uncmp, => v_comp_ratio, => v_comptype_str, => DBMS_COMPRESSION.COMP_RATIO_ALLROWS); JINE('Compression Type: ' TO_CHAR(v_comptype_str)); JINE('Estimated Compression Ratio: ' TO_CHAR(v_cmp_ratio));</pre>
<pre>comptype blkcnt_cmp blkcnt_uncmp row_cmp row_uncmp cmp_ratio comptype_str subset_numrows DBMS_OUTPUT.PUT_I DBMS_OUTPUT.PUT_I END;</pre>	<pre>=> DBMS_COMPRESSION.COMP_INMEMORY_QUERY_LOW, => v_blkcnt_cmp, => v_blkcnt_uncmp, => v_row_cmp, => v_row_uncmp, => v_cmp_ratio, => v_comptype_str, => DBMS_COMPRESSION.COMP_RATIO_ALLROWS); LINE('Compression Type: ' TO_CHAR(v_cmptype_str)); LINE('Estimated Compression Ratio: ' TO_CHAR(v_cmp_ratio));</pre>

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

Compression

ALTER MATERIALIZED VIEW mv1 INMEMORY MEMCOMPRESS FOR QUERY LOW;

```
CREATE TABLE trades
  (Name varchar(20),
   Desc varchar(200))
INMEMORY
MEMCOMPRESS FOR DML(desc);
```

- Objects compressed during population
- New compression techniques
 - Focused on scan performance
- 2x to 20x compression typical
- Multiple levels of compression
 - FOR DML
 - FOR QUERY LOW/HIGH
 - FOR CAPACITY LOW/HIGH
- Possible to use a different level for different partitions in a table

Columns Can Be Excluded

ALTER TABLE sales INMEMORY NO INMEMORY (delivery_note);

- You don't have to populate all columns
 - If excluded columns are accessed then the query will run against the row-store
 - In 21c, In-Memory Hybrid Scans supports both in-memory scans and row-store access for excluded projection 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

How Does Database In-Memory Work With RAC?

Parallel Query on RAC with Database In-Memory

- Scale-Out across servers to grow memory and CPUs
- Shared nothing architecture
- IMCUs not shipped across interconnect – cache fusion is not in play!
- In-Memory queries are parallelized across servers to access local columnar data

RAC : In-Memory and Distribution of Data

ALTER TABLE sales INMEMORY;

ALTER TABLE sales INMEMORY DISTRIBUTE BY PARTITION;

ALTER TABLE sales INMEMORY DISTRIBUTE ROWID RANGE;

- Distribution allows in memory segments larger than individual instance memory
- Policy is automatic (Distribute AUTO) or user-specifiable
- Controlled by DISTRIBUTE subclause
 - Distribute by rowid range
 - Distribute by partition
 - Distribute by subpartition
- Goal: Ensure Even Distribution

Querying In-Memory data in a RAC environment

- Shared nothing architecture means Parallel Query must be used to access data
- Must have a DOP greater than or equal to the number of column stores
- Query coordinator automatically starts parallel server processes on the correct nodes (Requires Auto DOP in 12.1.0.2)

How Does Database In-Memory Work With Other Database Features?

Why In-Memory on Exadata: Unique Features

Unique to Exadata

- In-Memory formats on Exadata Flash
- In-Memory Duplication
- In-Memory on Active Data Guard

Available on All Flavors of Exadata

- On-Premises
- Exadata Cloud Service
- Exadata Cloud at Customer

Database In-Memory Works (Better) with Multitenant

Multitenant consolidation "gives back" resources that DB In-Memory needs

Consolidation with Multitenant frees up

- Memory
- CPU Cycles

Oracle Database In-Memory wants

- Memory
- CPU Cycles

How Do I Tell If The In-Memory Column Store Is Being Used?

Target The Right Workloads

Understand Where In-Memory Helps

- In-Memory speeds up analytic data access, not:
 - DML (insert, update, delete)
 - Network round trips, logon/logoff
 - Parsing, PL/SQL, complex functions
 - Data processing (as opposed to access)
 - Complex joins or aggregations where not much data is filtered before processing
 - Load and select once Staging tables, ETL, temp tables
- Process data in sets of rows in the Database and not one row at a time in the application

Know your bottleneck!

Which Queries Benefit From Database In-Memory?

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

is spent accessing data

is spent joining data

is spent aggregating data

Use Time Based Analysis Techniques To Evaluate Benefit

< Overvie

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

				Time & Wait	Statistic	5		10	Stati	stics		
SQL Text SELECT /*+ N0_PARALLEL MONIT a Started Thu Jan 12, 2017 1:51:44 PM esh Time Thu Jan 12, 2017 1:51:45 PM cution ID 16777216 User SSB etch Calls 1	OR */ count(*), :	su [Duratio Database Tim PL/SQL & Jav Activity 9	n e e e e e e e e e e e e e e e e e e e		1.03 0.99	s B s IO	uffer G Reque IO B)	Gets ests 0 ytes 0		3
Statistics Plan 📐 Activity					-			-	-			
Operation	Name	Estimat	Cost	Timeline(1s)	Exec	Actual	Memor	Temp	0	IO Re	IO	Activity %
SELECT STATEMENT					1	1						
- SORT AGGREGATE		1			1	1						
TABLE ACCESS INMEMORY FULL	LINEORDER	10M	12K		1	10M						100
										Cpu: in	memory	1 samples (100%
	SQL Text SELECT /*+ NO_PARALLEL MONIT 1 Started Thu Jan 12, 2017 1:51:44 PM esh Time Thu Jan 12, 2017 1:51:45 PM cution ID 16777216 User SSB stch Calls 1 Statistics Image: Plan Mathematical Statistics Image: Plan Statistics Image: Plan Statistics Image: Plan User SSB store Plan Note Operation Image: Plan Image: SORT AGGREGATE Image: Table ACCESS INMEMORY FULL	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), 1 Started Thu Jan 12, 2017 1:51:44 PM esh Time Thu Jan 12, 2017 1:51:45 PM cution ID 16777216 User SSB stch Calls 1 Statistics	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU 1 Started Thu Jan 12, 2017 1:51:44 PM esh Time Thu Jan 12, 2017 1:51:45 PM cution ID 16777216 User SS8 stch Calls 1 Statistics	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU 1 Started Thu Jan 12, 2017 1:51:44 PM esh Time Thu Jan 12, 2017 1:51:45 PM cution ID 16777216 User SSB stch Calls 1 Statistics Image: Plan image: Activity sh Value 2267213921 Plan Note Plan Note Operation Name E- SORT AGGREGATE 1 Image: TABLE ACCESS INMEMORY FULL LINEORDER 10M	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU Duratio I Started Thu Jan 12, 2017 1:51:44 PM Database Tim esh Time Thu Jan 12, 2017 1:51:45 PM Database Tim cution ID 16777216 Activity % User SS8 Activity % statistics Time Plan Activity Statistics Time Plan Activity Statistics Timeline(1s) Statistics StateMathematic Cost Timeline(1s) Image: Sort AGGREGATE 1 Image: Table ACCESS INMEMORY FULL LINEORDER 10M 12K	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU In Started Thu Jan 12, 2017 1:51:44 PM esh Time Thu Jan 12, 2017 1:51:45 PM cution ID 16777216 User SSB stch Calls 1 Statistics Image: Plan image: Plan image: Plan Note Disperation Name Estimat Cost Timeline(1s) Exec Disperation Name Image: Plan image: Plan image: Plan image: Plan Note Image: Plan image: Plan image: Plan image: Plan Note Disperation Name Estimat Cost Timeline(1s) Exec Image: Plan image: Plan image: Plan image: Plan Note Image: Plan ima	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU 1 Started Thu Jan 12, 2017 1:51:44 PM esh Time Thu Jan 12, 2017 1:51:45 PM cution ID 16777216 User SSB stch Calls 1 Statistics	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU Duration 1.00 1 Started Thu Jan 12, 2017 1:51:44 PM 0.99 esh Time Thu Jan 12, 2017 1:51:45 PM 0.99 cution ID 16777216 0.90 User SSB 100 stch Calls 1 100 Statistics The Plan Activity sth Value 2267213921 Plan Note Operation Name Estimat Cost Timeline(1s) Exec Actual Memor E-SELECT STATEMENT 1 1 1 1 1 1 1 TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K 1 10M 1	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU Duration 1.0s 1.0s i Started Thu Jan 12, 2017 1:51:44 PM Database Time 0.9s PI ish Time Thu Jan 12, 2017 1:51:45 PM Database Time 0.9s PI cution ID 16777216 Detabase Time 0.9s PI User SSB	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU Duration 1.0s Buffer Q 1 Started Thu Jan 12, 2017 1:51:44 PM Duration 0.9s PL/SQL & Java 0s cution ID 16777216 0.9s 100 100 User SSB Statistics Sign Plan Activity	SQL Text SELECT /* + NO_PARALLEL MONITOR */ count(*), SU Duration 1.0s 1.0s in Started Thu Jan 12, 2017 1:51:44 PM Duration 0.se 0.se sth Time Thu Jan 12, 2017 1:51:45 PM Duration 0.se 0.se cution ID 16777216 100 100 100 User SS8 100 Activity % 100 10 Bytes 0 statistics Image: Single Plan Activity Image: Single Plan Activity Image: Single Plan <	SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU Duration 1.0s Buffer Gets IO ish Time Thu Jan 12, 2017 1:51:44 PM 0.9s PL/SQL & Java 0.s 0.0s IO Buffer Gets IO Buffer Gets IO Requests 0 IO Buffer Gets Buffer Gets Buffer Gets Buffer Gets Buffer Gets Buffer Gets

Use Time Based Analysis Techniques To Evaluate Benefit

Execution Started Thu Jan 12, 2017 1:51:44 PM

Last Refresh Time Thu Jan 12, 2017 1:51:45 PM

Plan Statistics Plan Activity

Execution ID 16777216

User SSB Fetch Calls 1

SQL Text SELECT /*+ NO_PARALLEL MONITOR */ count(*), SU

Time & Wait Statistics

Duration

Database Time

PL/SOL & Java Os

Activity %

....

IO Statistics

Buffer Gets

IO Requests 0

IO Bytes 0

Overview

General

Details

SQL Monitor Active Reports

 Shows how SQL was executed and where time was spent

	-											
Operation	Name	Estimat	Cost	Timeline(1s)	Exec	Actual	Memor	Temp	0	IO Re	IO	Activity %
□-SELECT STATEMENT				_	1	1						
SORT AGGREGATE		1		_	1	1						
TABLE ACCESS INMEMORY FULL	LINEORDER	10M	12K		1	10M						100
										Cpu: in r	nemory:	1 samples (100%
	- SELECT STATEMENT - SORT AGGREGATE - TABLE ACCESS INMEMORY FULL	SELECT STATEMENT SORT AGGREGATE TABLE ACCESS INMEMORY FULL LINEORDER	SELECT STATEMENT SORT AGGREGATE TABLE ACCESS INMEMORY FULL LINEORDER 10M	SELECT STATEMENT SORT AGGREGATE TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K	SELECT STATEMENT SORT AGGREGATE TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K	SELECT STATEMENT 1 SORT AGGREGATE 1 TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K	SELECT STATEMENT 1 SORT AGGREGATE 1 TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K 10M	SELECT STATEMENT 1 1 SORT AGGREGATE 1 1 TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K 1 10M	SELECT STATEMENT SORT AGGREGATE TABLE ACCESS INMEMORY FULL LINEORDER 10M 10K 10K 10K 10K 10K	SELECT STATEMENT I SORT AGGREGATE 1 TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K 1 10M 10 10M	SELECT STATEMENT I SORT AGGREGATE 1 TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K INDECREDER 10M 10M 10M INDECREDER 10M <	SELECT STATEMENT I SORT AGGREGATE 1 TABLE ACCESS INMEMORY FULL LINEORDER 10M 12K 1 10M 1 10M 1 10M

Tip: Identifying In-Memory Benefits

- Session level statistics
- Best way to determine if In-Memory was used
- Best way to measure the benefits of an In-Memory scan
- See blogs.oracle.com/inmemory for descriptions of the key statistics

IM scan bytes in-memory IM scan bytes uncompressed IM scan CUs columns accessed IM scan CUs columns decompressed IM scan CUs columns theoretical max IM scan rows IM scan rows range excluded IM scan rows excluded IM scan rows optimized IM scan rows projected IM scan CUs predicates received IM scan CUs predicates applied IM scan CUs predicates optimized IM scan CUs pruned IM scan segments minmax eligible

What's New?

Self Managing In-Memory

 In-Memory Spatial Analytics In-Memory Full Text Columns External Table Enhancements

Database In-Memory Innovations

12.1

- Pure In-Memory column format
- Scan & Filter on compressed data
- **Fast ioins**
- Data pruning via storage indexes
- SIMD vector processing
- **In-Memory aggregation**

2.1

- 12.2
- Join Groups
- In-Memory Expressions
- **JSON/OSON** support
- Massive capacity In-Memory on Exadata flash
- Auto population policies
- Fast-Start
- Active Data Guard

12.2

Automatic In-Memory

In-Memory Dynamic Scans

18c

- In-Memory External tables
- In-Memory Optimized Arithmetic
- Memoptimized Rowstore –
- Fast Lookup

18c

21c

 Hybrid Scans
 JSON Data Type Vector Joins
 Base Level Feature

> Self-Managing, Convergence

Performance, Automation

19c

External Tables: Hive

Rowstore – Fast Ingest

19c

Performance

Memoptimized

& HDFS

Performance, Capacity

IM Column Store With The Base Level Feature

- A 21c feature
- Now available in 19c with the July 19.8 Release Update
- Enabled with INMEMORY_FORCE parameter
 - Must be set to BASE_LEVEL
- INMEMORY_SIZE limited to a maximum of 16GB
- Feature tracking reports usage as "In-Memory Base Level"

Where Can You Get More Information?

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

O Oracle	Database In-Memory	Q =
Oracle D	atabase In-Memory	
	DBIM Resources	
	The second se	
	Andy threnes	
	August 27, 2021	
	Text Size (10%): +	
	About	
	About	
	Oracle Database In-Memory Resources page that lists helpful links to Database In-Memory information.	
	Technical Information	
	Oracle Database In-Memory Data Sheet Oracle Database In-Memory Data Sheet	
	Oracle Database in Memory Duick Start fone page brieft	
	Oracle Database In-Memory Quick Start Guide	
	Oracle Database In-Memory Implementation Guidelines Technical Brief	
	Oracle Database In-Memory: In-Memory Aggregation	
	When to Use Oracle Database In-Memory	
	Why Exadata is The Best Platform For Database In-Memory	
	Oracle Database In-Memory Advisor	
	 Benchmark Results Reveal the Benefits of Oracle Database in-Memory for SAP Applications 	
	Enterprise Data Architecture	