

@chandlerDBA http://chandlerDBA.com

Wisdom comes from experience. Experience is often a result of lack of wisdom.

STATISTICS

DOING IT RIGHT, THE EASY WAY

Neil Chandler Chandler Systems



Independent Database Consultant Working in IT for over 30 years [WTF!]



BLOG: http://chandlerdba.com Twitter: **@chandlerDBA** E: neil@chandler.uk.com

It's not worth doing something unless someone, somewhere, would much rather you weren't doing it.

VIDEO NOT IN PDF

@chandlerDBA http://chandlerDBA.com

A long time ago...

@chandlerDBA http://chandlerDBA.com

RULE BASED OPTIMISER (RBO)

- Deprecated in Oracle 10G, but still used by Oracle themselves 1.
- Didn't know anything about the shape of your data 2.
- Followed rules based upon how you coded your SQL, 3. plus some heuristics for the access path
- Probably the source of the "Full Table Scan is Bad" myth 4.
- 5. Still used (a little bit) by Oracle when accessing the Data Dictionary





COST-BASED OPTIMISER (CBO)

Diagram borrowed from Oracle 19c Tuning Guide

Contributing Authors: Nigel Bayliss, Maria Colgan, Tom Kyte



DBA TABLES DBA TAB STATISTICS DBA TAB PARTITIONS DBA TAB SUB PARTITIONS DBA TAB COLUMNS DBA TAB COL STATISTICS DBA PART COL STATISTICS DBA SUBPART COL STATISTICS DBA INDEXES DBA IND STATISTICS DBA IND PARTITIONS DBA IND SUBPARTIONS DBA TAB HISTOGRAMS DBA PART HISTOGRAMS DBA SUBPART HISTOGRAMS

Statistics

101100100

DBA_TAB_STATISTICS

OWNER Owner of the object TABLE NAME Name of the table Name of the partition PARTITION PARTIT Position of the partition within table CITION NAME of the subpartition SUBPAI Position f the subpartition within partition SUBPA TITION POSITION OBJE Type of the one of (TABLE, PARTITION, SUBPARTITION) The number of rows In the object NUM e number of use he number of empty blocks I The average available free space in the The number of chained rows in the object The average unrepace of all blocks on a f LOCKS The average unrepace of all blocks on a f The average unrepace of all blocks on a f The average unrepace of all blocks on a f the object BLO EMI AV C NUM TELIST BLOCKS AVG CACHED OCKS AVG_CACHE_HIT_RAL IM IMCU COUNT her of IMCUs in the object Number ... immemory blocks in the object IM BLOCK COUNT The timestamp net recent update to the inmemory stat IM STAT UPDATE TIME SCAN RATE Scan rate for the object The sample size used in analy ing this table SAMPLE SIZE The date of the most recent time the table was analyzed LAST ANALYZED Are the statistics calculated without mergan underlying par GLOBAL STATS itions? Were the statistics entered directly by the user. USER STATS STATTYPE LOCKED type of statistics lock STALE STATS Whether statistics for the object is stale or not NOTES Notes regarding special properties of the stats whether statistics for the object is shared or session SCOPE

@chandlerDBA http://chandlerDBA.com

STATISTICS. DOING IT RIGHT, THE EASY WAY

Statistics

101100100

DBA_TAB_STATISTICS



NUM_ROWS	The number of rows in the object = relative scale
BLOCKS	The number of used blocks in the object = I/O
SAMPLE_SIZE	The sample size used in stats gather
LAST_ANALYZED	The date of the most recent time this table
	was analyzed
GLOBAL_STATS	Are the statistics calculated
	without merging underlying partitions?
USER_STATS	Were the stats entered directly by the user?
STATTYPE_LOCKED	Locked stats aren't gathered automatically
STALE_STATS	<pre>%change - DBA_TAB_MODIFICATIONS (FLUSH first!)</pre>
NOTES	New for 19C
	Notes regarding special properties of the stats

DBA_TAB_COL_STATISTICS





@chandlerDBA http://chandlerDBA.com

DBA_TAB_COL_STATISTICS



NUM DISTINCT The number of distinct values in the column The lowest value in the column LOW VALUE HIGH VALUE The highest value in the column Query costs lowered for query outside of these values 1/NUM DISTINCT = % of table retrieved for DENSITY **any specific value** (ignored if you have histogram) The number of nulls in the column NUM NULLS The sample size used in analyzing this column SAMPLE SIZE HISTOGRAM What type of histogram on the column NUM BUCKETS The number of buckets in histogram for the column

DBA IND STATISTICS



NUM ROWS LEAF BLOCKS CLUSTERING FACTOR

The number of rows in the index = SCALE The number of leaf blocks in the index = I/ODISTINCT KEYS Derive % of index to be used (NUM ROWS/DISTINCT KEYS) AVG LEAF BLOCKS PER KEY Effectively LEAF BLOCKS/DISTINCT KEYS AVG DATA BLOCKS PER KEY Effectively CLUSTERING FACTOR/DISTINCT KEYS How aligned is the index to the table? Can we re-read the **same** table block for the next index value? [If "NO" Increment the CF]

DBA_IND_STATISTICS

1 0 1 1 0 0 1 0 0 Statistics

CLUSTERING FACTOR:





CLUSTERING_FACTOR: 2

CLUSTERING_FACTOR: 5

@chandlerDBA http://chandlerDBA.com

DBA IND STATISTICS



NUM ROWS LEAF BLOCKS CLUSTERING FACTOR

The number of rows in the index = SCALE The number of leaf blocks in the index = I/ODISTINCT KEYS Derive % of index to be used (NUM ROWS/DISTINCT KEYS) AVG LEAF BLOCKS PER KEY Effectively LEAF BLOCKS/DISTINCT KEYS AVG DATA BLOCKS PER KEY Effectively CLUSTERING FACTOR/DISTINCT KEYS How aligned is the index to the **table?** Can we re-read the table block for the next index value? If CF is close to **#table-blocks=good index** If CF is close to **#table-***rows* =**no-so-good index** Can only be changed by reorganizing and reordering the **table** Default is too pessimistic - set TBC to 16! From 12.1 (or patched 11.1/11.2 PatchID is 15830250)

TABLE CACHED BLOCKS stats gather option makes this number more realistic

DBA_TAB_HISTOGRAMS



Statistics

101100100

DBA_TAB_HISTOGRAMS



DBA_TAB_COLUMNS

Table ColumnHistogram #BucketsLow_ValHigh_ValDecode_LowDecode_HighSALESCHANNEL_IDNONE1C103C10A29

HISTOGRAM:	NONE !			
COLUMN	ENDPOINT	NUMBER	ENDPOINT	VALUE
CHANNEL_ID	0		2	_
CHANNEL ID	1		9	

@chandlerDBA http://chandlerDBA.com

STATISTICS. DOING IT RIGHT, THE EASY WAY

DBA_TAB_HISTOGRAMS



DBA_TAB_COLUMNS

Table	Column	Histogram #1	Buckets	Low_Val	High_Val	Decode_Low	Decode_High
SALES	CHANNEL_ID	NONE	1	C103	C10A	2	9
SALES	PROMO_ID	FREQUENCY	4	C122	C20A64	33	999

HISTOGRAM:	FREQUENCY						
COLUMN	ENDPOINT_NUMBER	VALUE	Maths				
	0						
PROMO ID	2074	33	2074- 0) =	2074	Х	33
PROMO_ID	20096	350	20096- 2074	=	18022	Х	350 - 1%-index?
PROMO_ID	31006	351	31006-20096	5 =	10910	Х	351
PROMO_ID	918843	999	918843-31006	5 =	887837	Х	999 🗲 96% - FTS

@chandlerDBA http://chandlerDBA.com

DBA_TAB_HISTOGRAMS

1 0 1 1 0 0 1 0 0 Statistics

DBA_TAB_COLUMNS Table Column SALES CHANNEL_ID SALES PROMO_ID SALES AMOUNT_SOLD	Histogram NONE FREQUENCY HYBRID	#Buckets 1 4 254	Low_Val C103 C122 C10729	High_Val C10A C20A64 C2125349	Dec 2 33 6.4	sca num_rov to get a	ale up by vs/sample_size approx count
HISTOGRAM: HY	BRID						
COLUMN ENI	DP_# VALUE	E ENDPOI	NT_REPE	AT_COUNT]	Maths	
AMOUNT_SOLD	1 6.40	C		9	:	_10=	9 x 6.40
AMOUNT_SOLD	33 7.10	C		19	:	33-1=32	19 x 7.10
AMOUNT SOLD	56 7.32	1		4	:	56-33=23	4 x 7.31
AMOUNT_SOLD <i>etc</i>	78 7.44	1		3		78-56-22	3 x 7.44

@chandlerDBA http://chandlerDBA.com

STRATEGIES

- **Oracle "Internal" Statistics**
- **Dictionary Stats**
- Oracle 12+ gathers dictionary stats automatically
- Gather them yourself when your schemas are in place
- Re-gather if you make significant change
- Re-gather before an upgrade
- Re-gather after an upgrade



exec DBMS_STATS.GATHER_DICTIONARY_STATS;

STRATEGIES

Oracle "Internal" Statistics

Fixed Object Stats (X\$ tables)

Oracle 12+ gathers **missing** stats automatically at the **end** of the maintenance window *(if there's time)*

- Gather them **yourself** during a *representative* workload
- Re-gather if you make changes to instance structure, such as SGA size, or workload changes

exec DBMS_STATS.DELETE_FIXED_OBJECT_STATS; exec DBMS_STATS.GATHER_FIXED_OBJECTS_STATS;



SELECT *
FROM dba_tab_statistics
WHERE object_type = 'FIXED TABLE'

STRATEGIES

- **Oracle** "Internal" Statistics
- **System Statistics**
- Measures your CPU and storage capabilities
- Should we be gathering systems statistics?



(other opinions are available)



**unless you are using a dedicated Exadata for true Data Warehouse loads

YOUR STATISTICS

The Gathering



@chandlerDBA http://chandlerDBA.com

YOUR STATISTICS

Automatic Statistics Gathering



@chandlerDBA http://chandlerDBA.com

REAL WORLD

BATCH CODE in 12.2 database (written *long ago* in Oracle 10G):

19:00 load data

20:00 dbms_stats.gather_table_stats(USER,'TABLE_X',ESTIMATE_PERCENT=>10,...)

21:00 perform data manipulation with lovely new stats

22:00 Autotask job does the gather again (stale dependent), with different (default) options

04:00 A manual job starts and gathers the stats again.

They had disabled the Autotask job... an upgrade re-enabled it.

Key Questions

1. How do we ensure it's always the same gather?

2. How do we avoid repeating work?

3. Do I have to re-write all of my batch?

BATCH CODE in 12.2 database (written *long ago* in Oracle 10G): 19:00 load data

- 20:00 dbms_stats.gather_table_stats(USER,'TABLE_X',ESTIMATE_PERCENT=>10,.
- 21:00 perform data manipulation with lovely new stats
- 22:00 Autotask job does the gather again (stale dependent), with different (default) 04:00 A manual job starts and gathers the stats again.

1. How do we ensure it's always the same gather?

Don't specify options on the command line: use TABLE PREFS

```
DBMS_STATS.SET_TABLE_PREFS
(user,'TABLE_X',
'ESTIMATE PERCENT',DBMS STATS.AUTO SAMPLE SIZE)
```

BATCH CODE in 12.2 database (written *long ago* in Oracle 10G):

19:00 load data

- 20:00 dbms_stats.gather_table_stats(USER,'TABLE_X',ESTIMATE_PERCENT=>10,...)
- 21:00 perform data manipulation with lovely new stats
- 22:00 Autotask job does the gather again (stale dependent), with different (default) options 04:00 A manual job starts and gathers the stats again.

2. How do we avoid repeating work?

Adjust Windows? DBMS_SCHEDULER.set_attribute Disable Autotask & start it yourself? DBMS_AUTO_TASK_IMMEDIATE.GATHER_OPTIMIZER_STATS

```
Maybe adjust the STALE percent for that table?
DBMS_STATS.SET_TABLE_PREFS
(user, 'TABLE_X', 'STALE_PERCENT', '50')
```



BATCH CODE in 12.2 database (written *long ago* in Oracle 10G): 19:00 load data

19:00 load data

20:00 dbms_stats.gather_table_stats(USER,'TABLE_X',ESTIMATE_PERCENT=>10,...)

21:00 perform data manipulation with lovely new stats

22:00 Autotask job does the gather again (stale dependent), with different (default) options 04:00 A manual job starts and gathers the stats again.

3. Do I have to re-write lots of my code?

From 12.2 we can OVERRIDE command-line options, forcing the stats gather to use TABLE PREFS

DBMS_STATS.SET_TABLE_PREFS

(user, 'TABLE_X', 'PREFERENCE_OVERRIDES_PARAMETER', 'TRUE')

SELECT * FROM DBA_TAB_STAT_PREFS

TABLE PREFERENCE NAME

CUSTOMERS METHOD OPT

PREFERENCE_VALUE
FOR ALL COLUMNS SIZE 1
FOR COLUMNS SIZE 1000 COUNTRY_ID,CUST_ID
FOR COLUMNS SIZE AUTO CUST_GENDER
.000000

SALES ESTIMATE PERCENT

SALES PREFERENCE OVERRIDES PARAMETER TRUE



REAL-WORLD : GLOBAL_PREFS

You can also set preferences at a GLOBAL level:

exec DBMS STATS.SET GLOBAL PREFS ('METHOD OPT', 'FOR ALL COLUMNS SIZE 1');



GLOBAL PREFS in 19C (12 prefs in 11G, 14 new in 12C/18C/19C) select dbms stats.get prefs('CASCADE') from dual; select dbms stats.get prefs('DEGREE') from dual; select dbms stats.get prefs('ESTIMATE PERCENT') from dual; select dbms stats.get prefs('METHOD OPT') from dual; select dbms stats.get prefs('NO INVALIDATE') from dual; select dbms stats.get prefs('GRANULARITY') from dual; select dbms stats.get prefs('PUBLISH') from dual; select dbms stats.get prefs('INCREMENTAL') from dual; select dbms stats.get prefs('INCREMENTAL LEVEL') from dual; select dbms stats.get prefs('STALE PERCENT') from dual; select dbms stats.get prefs('AUTOSTATS TARGET') from dual; select dbms stats.get prefs('CONCURRENT') from dual; select dbms stats.get prefs('INCREMENTAL STALENESS') from dual; select dbms stats.get prefs('GLOBAL TEMP TABLE STATS') from dual; select dbms_stats.get_prefs('TABLE_CACHED_BLOCKS') from dual; select dbms stats.get prefs('OPTIONS') from dual; select dbms stats.get prefs('STAT CATEGORY') from dual; select dbms_stats.get_prefs('PREFERENCE_OVERRIDES_PARAMETER') from dual; select dbms stats.get prefs('APPROXIMATE NDV ALGORITHM') from dual; select dbms_stats.get_prefs('AUTO_STAT_EXTENSIONS') from dual; select dbms stats.get prefs('WAIT TIME TO UPDATE STATS') from dual; select dbms stats.get prefs('ROOT TRIGGER PDB') from dual; select dbms stats.get prefs('COORDINATOR TRIGGER SHARD') from dual; select dbms stats.get prefs('AUTO TASK STATUS') from dual; select dbms stats.get prefs('AUTO TASK MAX RUN TIME') from dual; select dbms stats.get prefs('AUTO TASK INTERVAL') from dual;

STATISTICS. DOING IT RIGHT, THE EASY WAY

@chandlerDBA http://chandlerDBA.com

DBA AUTOTASK JOB HISTORY

JOB STATUS: STOPPED

JOB_INFO : REASON="Stop job called because associated window was closed"

Options: Make the window bigger or **speed it up**!



@chandlerDBA http://chandlerDBA.com

Speed It Up!!!

Parallelise the stats gather for a large table!

```
DBMS_STATS.SET_TABLE_PREFS
('SCHEMA', 'TABLE X', 'DEGREE', 4)
```

Go insane and let Oracle parallelise all of it!

```
DBMS_STATS.SET_GLOBAL_PREFS
('DEGREE', DBMS STATS.AUTO DEGREE)
```

Gather Statistics Concurrently - several tables at the same time...

DBMS_STATS.SET_GLOBAL_PREFS('CONCURRENT','AUTOMATIC')

MANUAL: Enabled only for manual statistics gathering AUTOMATIC: Enabled only for the auto statistics gathering ALL: Enabled for all statistics gathering calls OFF: Concurrency is disabled (default)

Starts many Scheduler Jobs simultaneously!

Also need the following privileges: CREATE JOB, MANAGE SCHEDULER, MANAGE ANY QUEUE



Speed It Up!!!

Speed It Up!!!

Parallel or Concurrent (or both) gathering of stats means you should use Resource Manager to limit the total resources consumed

DBA_RSRC_CONSUMER_GROUPS: ORA\$AUTOTASK

If you are doing ANY PARALLEL processing you *really* should use **Resource Manager** to control it!



STATISTICS. DOING IT RIGHT, THE EASY WAY

@chandlerDBA http://chandlerDBA.com

Resource Manager

```
begin
dbms resource manager.clear pending area();
dbms resource manager.create pending area();
dbms resource manager.update plan directive(
                            => 'DEFAULT MAINTENANCE PLAN',
 plan
 group or subplan => 'ORA$AUTOTASK',
 new mgmt pl
            => 10,
                                              -- Gets at least 10%
 new max utilization limit => 50, -- Not allowed more than 50% (if the CPU is busy)
 new parallel degree limit p1 => 4); -- And don't go wild...
 dbms resource manager.update plan directive(
                            => 'DEFAULT MAINTENANCE PLAN',
 plan
 group_or_subplan
                            => 'OTHER GROUPS',
 new mgmt pl
                            => 20);
 dbms_resource_manager.update_plan_directive(
 plan
                            => 'DEFAULT MAINTENANCE PLAN',
 group or subplan
                            => 'SYS GROUP',
                                                 select ... from DBA RSRC PLAN DIRECTIVES
                            => 70);
 new mgmt pl
 dbms resource manager.validate pending area;
                                                 where plan = 'DEFAULT MAINTENANCE PLAN';
 dbms resource manager.submit pending area;
end;
                                                 GROUP OR SUBPLAN MGMT P1 PARALLEL DEGREE_LIMIT_P1 MAX_U
                                                 ORA$AUTOTASK
                                                                                               50
                                                                     10
                                                 OTHER GROUPS
                                                                     20
                                                 SYS GROUP
                                                                     70
```

MANUALLY SETTING STATISTICS

DBMS_STATS.SET_TABLE_STATS

- 1. Set your stats
- 2. Lock the stats

3. Bask in the glow of consistent good plans?

REAL WORLD

The client had an unusual "small" database which was only accessed via IOT's (i.e. no humans with unexpected inputs)

Got some good stats via calculations + (some) experiments It was an unusual DB though. We also:

- disabled Hash Joins (NL was King here)
- NOARCHIVELOG mode
- Did NO backups

Locked it. Left it. Never gathered another statistic.

MANUALLY SETTING STATISTICS

DBMS_STATS.SET_TABLE_STATS

REAL WORLD EXAMPLE#2

This is actual code from a client. Did you set everything you need to set?

```
DBMS_STATS.SET_TABLE_STATS (
    ownname => 'SCHEMA',
    tabname => 'BIG_PART_TAB',
    numrows => l_numrows,
    numblks => l_num_blocks,
    avgrlen => l_avgrow_len,
    no_invalidate => TRUE,
    force => gather_force
    );
```

The input values for this are calculated from the (gathered) partitions.

@chandlerDBA http://chandlerDBA.com

MANUALLY SETTING STATISTICS



@chandlerDBA http://chandlerDBA.com

THOSE AREN'T THE STATS YOU'RE LOOKING DOR AT

Statistical (formerly Cardinality) Feedback



THOSE AREN'T THE STATS YOU'RE LOOKING DOR AT

Statistical (formerly Cardinality) Feedback

You run some SQL, it may identify different statistical values:

The STATISTICS COLLECTOR results can be seen in: **V\$SQL_PLAN_STATISTICS_ALL**

OPERATION	OPTIONS	O-NAME	O-TYP	E C	ARDINALITY	OUTPUT	ROWS
SELECT STATEMENT							2550
HASH JOIN					25		2550
NESTED LOOPS					25		5050
NESTED LOOPS					25		5050
STATISTICS COLLECT	OR						5050
TABLE ACCESS	FULL	TAB2	TABLE		25		5050
INDEX	UNIQUE SCAN	TAB3_PK	INDEX	(UNIQUE)	1		0
TABLE ACCESS	BY INDEX ROWID	TAB3	TABLE		1		0
TABLE ACCESS	FULL	TAB3	TABLE		1		7500

These statistics are only held in the SGA and will age out ...

THOSE AREN'T THE STATS YOU'RE LOOKING FOR AT

Dynamic Statistics (Sampling) - optimizer_dynamic_sampling

Level	When does it use Dynamic Stats?
0	off
1	if No stats on an unpartitioned table, and no indexes, and table is bigger than 32 blocks (samples 32 blocks)
2 (default)	if you have no stats on 1 table in the join or (and this is badly documented) if you use PARALLEL (samples 64 blocks)
3	(as 2) + if you have a complex predicate expression [e.g WHERE SUBSTR(column,1,3) =]
4	(as 3) + an OR or AND between multiple predicates on the same table
5-10	(as 4) but sample 128/256/512/1024/4096/ALL table blocks
11	Automatically Determined by Oracle

If you have a 10,000,000,000 row table, how representative is a 4096 block sample? If you are looking for consistent plans (i.e. consistent stats for OLTP), should you use this?

@chandlerDBA http://chandlerDBA.com

THOSE AREN'T THE STATS YOU'RE LOOKING FOR AT

Dynamic Statistics (Sampling) - optimizer dynamic sampling

In the **Notes** section of DBMS_XPLAN execution plan output it will show the level used:

Note

- dynamic statistics used: dynamic sampling (level=7) or
- dynamic statistics used for this statement (level=4)

The level will vary depending upon data size if using PARALLEL query from level 2.

In V\$SQL.SQL_TEXT you will see the dynamic sampling which is being executed:

Where your stats aren't good enough or are missing: SELECT /* OPT_DYN_SAMP */ ... SAMPLE BLOCK (0.51390, 8) SEED(1) "TABLE_X"

Where Dynamic Sampling/SQL Plan Directive has kicked in: SELECT /* DS_SVC */ ... SAMPLE BLOCK (0.51398, 8) SEED(1) "TABLE_X"

@chandlerDBA http://chandlerDBA.com

THOSE AREN'T THE STATS YOU'RE LOOKING DOR AT

Adaptive Statistics

- 12.1 Introduced and Enabled this by default
- 12.2 Retained the functionality but **Disabled** this feature by default optimizer_adaptive_statistics=FALSE

- What it does?
- Identify one or more COLUMNS with poor statistics (via Statistical Feedback)
- Create a SQL_PLAN_DIRECTIVE to perform Dynamic Sampling against those table column(s) (12.2+ also has a SQL_PLAN_DIRECTIVE to store Dynamically Sampled Stats)

To control in 12.1, see

"Recommendations for Adaptive Features in Oracle Database 12c Release 1 (12.1): (Doc ID 2187449.1)"





@chandlerDBA http://chandlerDBA.com

Statistics Advisor

- Oracle 12.2 comes with a statistics advisor to help you
- It runs nightly (window permitting): AUTO_STATS_ADVISOR_TASK or you can run it yourself

```
DECLARE
    task name VARCHAR2(128) := 'stats advisor report task';
    exec_name VARCHAR2(128) := NULL;
    report CLOB := NULL;
BEGIN
  -- create a task
    task name := dbms stats.create advisor task(task name);
  -- execute the task
    exec name := dbms stats.execute advisor task(task name);
  -- view the task report
    report := dbms stats.report advisor task (task name);
    dbms output.put line(report);
  -- for the Brave or Foolish ... implement the recommendation from the task
  --implementation result := dbms stats.implement advisor task(tname);
END;
```

Statistics Advisor

- Oracle 12.2 comes with a statistics advisor to help you
- It runs nightly (window permitting): AUTO_STATS_ADVISOR_TASK or you can run it yourself

FINDINGS

Rule Name:	UseConcurrent
Rule Description:	Use Concurrent preference for Statistics Collection
Finding:	The CONCURRENT preference is not used.
Recommendation:	Set the CONCURRENT preference.
Example:	dbms stats.set global prefs('CONCURRENT', 'ALL');
Rationale:	The system's condition satisfies the use of concurrent statistics
	gathering. Using CONCURRENT increases the efficiency of statistics
	gathering.

Statistics Advisor

- Oracle 12.2 comes with a statistics advisor to help you
- It runs nightly (window permitting): AUTO_STATS_ADVISOR_TASK or you can run it yourself

FINDINGS

Rule Name:	UseDefaultPreference
Rule Description:	Use Default Preference for Stats Collection
Finding:	Global preference METHOD OPT is set to a non-default value
	'FOR ALL COLUMNS SIZE 1'.
Recommendation:	Set the value of preference METHOD OPT to 'FOR ALL COLUMNS SIZE AUTO'.
Example:	Setting preference cascade to default value:
	dbms stats.set global prefs('CASCADE', NULL);
Rationale:	METHOD OPT controls the creation of histograms during statistics collection.
	With the default value FOR ALL COLUMNS SIZE AUTO, Oracle determines which
	columns require histograms and the number of buckets to use based on the usage
	of columns in SQL statements and the number of distinct values. The default
	value helps to create the necessary histograms with an adequate number of
	buckets.

@chandlerDBA http://chandlerDBA.com

Statistics Advisor: V\$STATS ADVISOR RULES;

ID	NAME	RULE_TYPE	DESCRIPTION
1	UseAutoJob	SYSTEM	Use Auto Job for Statistics Collection
2	CompleteAutoJob	SYSTEM	Auto Statistics Gather Job should complete successfully
3	MaintainStatsHistory	SYSTEM	Maintain Statistics History
4	UseConcurrent	SYSTEM	Use Concurrent preference for Statistics Collection
5	UseDefaultPreference	SYSTEM	Use Default Preference for Stats Collection
6	TurnOnSQLPlanDirective	SYSTEM	SQL Plan Directives should not be disabled
7	AvoidSetProcedures	OPERATION	Avoid Set Statistics Procedures
8	UseDefaultParams	OPERATION	Use Default Parameters in Statistics Collection Procedures
9	UseGatherSchemaStats	OPERATION	Use gather_schema_stats procedure
10	AvoidInefficientStatsOprSeq	OPERATION	Avoid inefficient statistics operation sequences
11	AvoidUnnecessaryStatsCollection	OBJECT	Avoid unnecessary statistics collection
12	AvoidStaleStats	OBJECT	Avoid objects with stale or no statistics
23	AvoidAnalyzeTable	OBJECT	Avoid using analyze table commands for statistics collection

Rules can be filtered to avoid repeatedly reporting against specific settings you have made: DBMS_STATS.CONFIGURE_ADVISOR_FILTER

190

2 New Statistics Features

Exadata and Cloud ONLY

- Real-Time Statistics
- High-Frequency Automatic Optimizer Statistics Collection

Table 1-8 Performance

Feature / Option / Pack	SE2	EE	EE- ES	DBCS SE	DBCS EE	DBCS EE-HP	DBCS EE-EP	ExaCS	Notes
Real-Time Statistics	N	N	Y	Y	Y	Y	Y	Y	EE-ES : Available on Exadata. Not available on Oracle Database Appliance.
High-Frequency Automatic Optimizer Statistics Collection	N	N	Y	Y	Y	Y	Y	Y	EE-ES : Available on Exadata. Not available on Oracle Database Appliance.

@chandlerDBA http://chandlerDBA.com

19C

Real-Time Statistics

- When a DML operation is currently modifying a table, the DB dynamically computes values for "the most essential statistics." (i.e. the easy ones)
 - LOW_VALUE
 - HIGH_VALUE
 - NUM_ROWS
- Reduces risk from "statistical decay & high-low value threat"
- Does it during the DML into memory buffers and flushes to disk occasionally (DBMS_STATS.FLUSH_DATABASE_MONITORING_INFO)
- Runs at an approximate 1% sample size to minimise impact to DML
- Check DBA_TAB_STATISTICS.NOTES

DBA_TAB_COL_STATISTICS.NOTES for value "**STATS_ON_CONVENTIONAL_DML**"

You still need to do traditional stats gathering!

190

High-Frequency Automatic Optimizer Statistics Collection

- Gathers "STALE" statistics much more frequently. Doesn't to any of the other autotask stats gathering stuff (e.g. Internal Stats, Stats Advisor)
- Does not replace AUTOTASK stats gathering
- Will not run during the auto stats gather window
- 3 new GLOBAL_PREFS
 - AUTO_TASK_STATUS ('OFF')
 - AUTO_TASK_MAX_RUN_TIME (3600 seconds)
 - AUTO_TASK_INTERVAL (run every 900 seconds. Min 60 seconds)
- Can see the runs in: DBA_AUTO_STAT_EXECUTIONS where ORIGIN='HIGH_FREQ_AUTO_TASK'

WHAT DIDN'T I COVER?

- Incremental Statistics for Partitioned Tables
- Synopses
- Partition Table Strategies using Copy
- Lots on Histograms
- Statistics History & Rolling Back Stats to previous versions
- Pending Statistics to test plans before you unleash them
- Undocumented Parameters
- and lots of other stuff...

COST-BASED OPTIMISER (CBO)

Important Quotes by Performance Experts



"You don't necessarily need up to date statistics. You need statistics that are representative of your data." - *Graham Wood*

"Do you want the optimizer to give you? The best performance, or consistent performance?" - *Anjo Kolk*

"You need to ensure that your stats tell Oracle what you want it to think your data looks like" - Jonathan Lewis

Time to use The Force Questions?

BLOG: http://chandlerdba.com Twitter: **@chandlerDBA** E: neil@chandler.uk.com

STATISTICS. DOING IT RIGHT, THE EASY WAY

@chandlerDBA http://chandlerDBA.com