Incremental Statistics: From a Nightmare to a Fast Solution

POUG 2018, 7 September 2018

Dani Schnider, Trivadis AG

@dani_schnider danischnider.wordpress.com

Dani Schnider

- Working for Trivadis in Glattbrugg/Zurich
  - Senior Principal Consultant
  - Data Warehouse Lead Architect
  - Trainer of several Courses
- Co-Author of the books
  - Data Warehousing mit Oracle
  - Data Warehouse Blueprints
- Certified Data Vault Data Modeler
- Oracle ACE

@dani_schnider

danischnider.wordpress.com

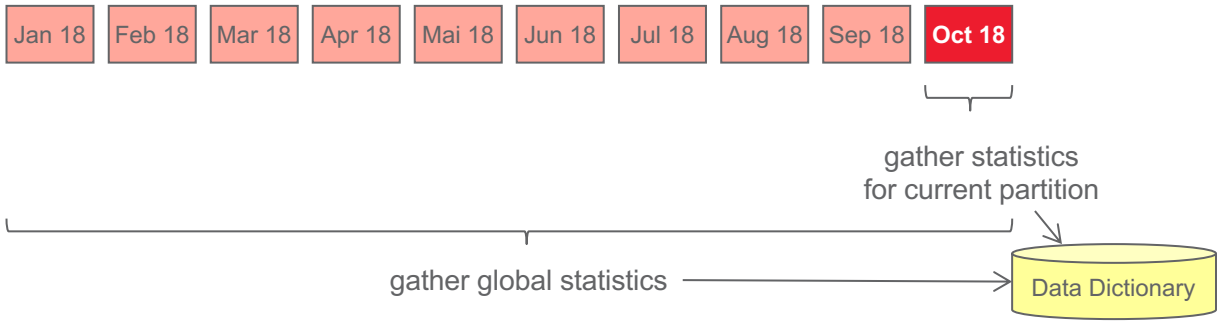
trivadis makes IT easier.

Incremental Statistics

trivadis
makes IT easier.

Global Statistics

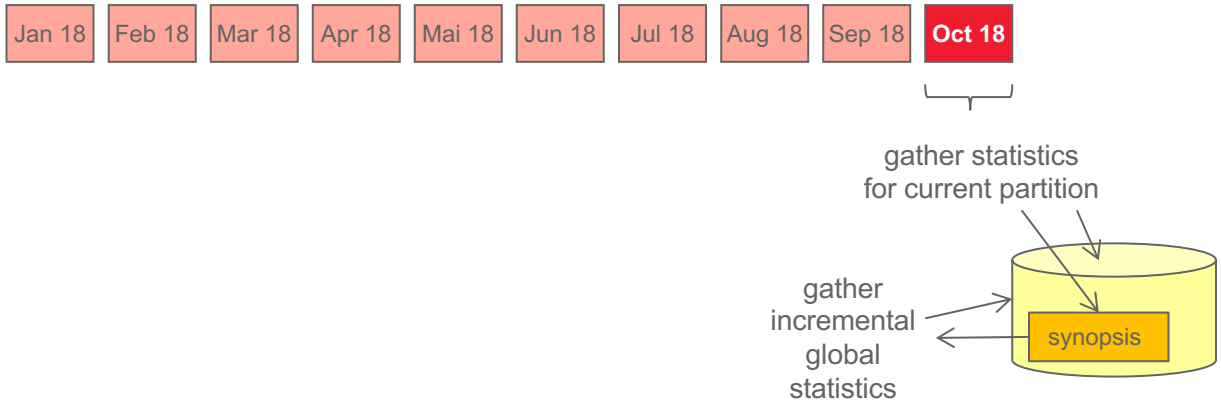
- Global statistics are essential for good execution plans
  - **num_distinct, low_value, high_value, density, histograms**
- Gathering global statistics is time-consuming
  - All partitions must be scanned



trivadis
makes IT easier.

Incremental Global Statistics

- Synopsis-based gathering of statistics (since Oracle 11g)
- For each partition a synopsis is stored in SYSAUX tablespace
  - Statistics metadata for partition and columns of partition
- Global statistics by aggregating the synopses from each partition



trivadis
makes IT easier.

Challenges of Incremental Statistics

- Only in combination with AUTO_SAMPLE_SIZE

```
dbms_stats.gather_table_stats
(ownname => USER, tabname => 'FCT_SALES', partname => 'PT_2018_01'
,estimate_percent => dbms_stats.auto_sample_size
,granularity => 'GLOBAL AND PARTITION');
```

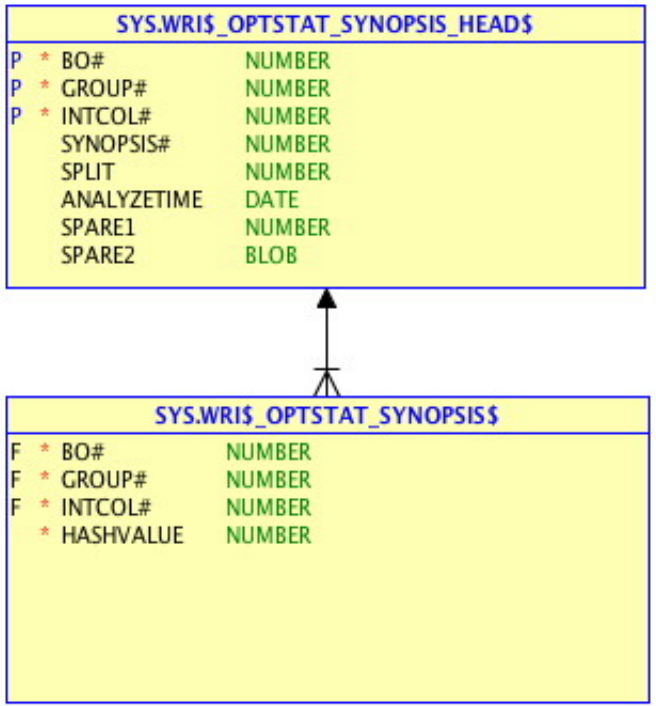
- High space requirements in tablespace SYSAUX
- Poor performance for databases with many partitions

trivadis makes IT easier.

Synopsis Tables

Two tables in SYSAUX tablespace:

- WRI$_OPTSTAT_SYNOPSIS_HEAD$
  - One row per table/partition/column
- WRI$_OPTSTAT_SYNOPSIS$
  - One row per distinct value per table/column
  - Since Oracle 12.1, this table is LIST-HASH partitioned per table(BO#) and partition (GROUP#)
  - In Oracle 12.2 and 18c, this table is only used for backward compatibility (see later)



trivadis makes IT easier.

A Real World Scenario

trivadis
makes IT easier.

Some Key Figures of Data Warehouse of a Customer

https://danischnider.wordpress.com/2017/12/30/incremental-statistics-a-real-world-scenario/

- Oracle Database 12.1.0.2 Enterprise Edition on Exadata
- Data Warehouse with around 24 TB of data
- 891 partitioned tables, most of them with INTERVAL daily partitions
- Total of 325851 partitions (state end of November 2017)

Synopsis Table	Rows	Size
WRI$_OPTSTAT_SYNOPSIS_HEAD$	13315001	718 MB
WRI$_OPTSTAT_SYNOPSIS$	9998263744	320 GB

trivadis makes IT easier.

Gathering Incremental Statistics

- Gathering statistics with DBMS_STATS is very slow
- This query was always on the Top Activity list of SQL statements

```
SELECT DISTINCT BO#, GROUP#
FROM SYS.WRI$_OPTSTAT_SYNOPSIS_HEAD$ H
WHERE GROUP#   <> 0
AND GROUP# NOT IN
   (SELECT T.OBJ# * 2 FROM SYS.TABPART$ T
   UNION ALL
   SELECT T.OBJ# * 2 FROM SYS.TABCOMPART$ T
   )
```

trivadis
makes IT easier.

Bugs, Bugs, Bugs

Examples:

- **23100700** (13-OCT-2017) Patch 23100700: PERFORMANCE ISSUE WITH RECLAIM_SYNOPSIS_SPACE
- **19450139** (01-NOV-2017) Patch 19450139: KN:LNX:PERFORMA… ISSUE WHEN RUNNIN… TABLE STAT… STA…
- **206027…** 20602794 … STATS GATHER R… …TEDLY GATHERING ON SAME PARTITIONS

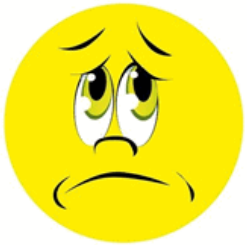
DETAILS

Consider the following known bugs when using incremental statistics:

Fixed 12.2.0.1 (Base Release)	Fixed 12.1.0.2 (Server Patch S…	…d
Patch:19790972 For 11.2.0.3,11.2.0.4,12.2.0.2, 12.1.0.2.160119ProactiveBP, 12.1.0.213 Proactive BP, 1.2.1.0.2.7Proactive BP		…9790972.8 …CAUSING LIBRARY CACHE …SUBPARTITION TABLE
Patch:16470836 For 11.2.0.4, 11.2.0.3,11.2.0.4171017		…836.8 …E FOR MON_MODS$ …D SQLS USING DBMS_STATS
Patch:17715578 … 11.2.0.3,11…	…4.170718	Document 17715578.8 HIGH CONCURRENT DBMS_STATS.GATHER_TABLE_STATS CAUSE CONTENTION ON COL_USAGE$
	…BP	Document 16851194.8 CONTINUED GROWTH IN SYSAUX WITH INCREMENTAL STATS WITHOUT GROWTH IN TABLE DATA **Note that this fix has been superseded by the fix in Bug:23100700**
…atch:20602794 for 11.2.0.4.0,12.1.0.2.1, 12.1.0.2.160419, 12.1.0.2.161018,12.1.0.2.17011,12.1.0.1.171017 Proactiv BP,12.1.0.1.170118 Proactiv BP		…THER …S ON SAME
	Patch:14461932 ,12.1.0.1.171017 Proactiv BP	…CS DOESN'T …RTITION …iew
Patch:14754390 for 11.2.0.4.12 Exadata, 11.2.0.4, 11.2.0.3	Patch:14754390 … Exadata,11.2.0.4, 11.2.0.3,	…OBJECT ENTRIES IN

My Oracle Support, Doc ID 2107602.1:
Things to Consider When Using Incremental Statistics

Our Solution on Oracle 12.1



- Disable Incremental Statistics
- Weekly job every Sunday to gather global statistics
- Waiting for Oracle 12.2 or Oracle 18c

trivadis makes IT easier.

Synopsis Calculation

trivadis
makes IT easier.

Synopsis Calculation (≤ Oracle 12.1)

- Synopsis calculation with **Adaptive Sampling**
- Number of distinct values (NDV) stored in WRI$_OPTSTAT_SYNOPSIS$
- Size of synopsis tables:
  - WRI$_OPTSTAT_SYNOPSIS_HEAD$

$$rows = number\ of\ partitions\ * number\ of\ columns$$

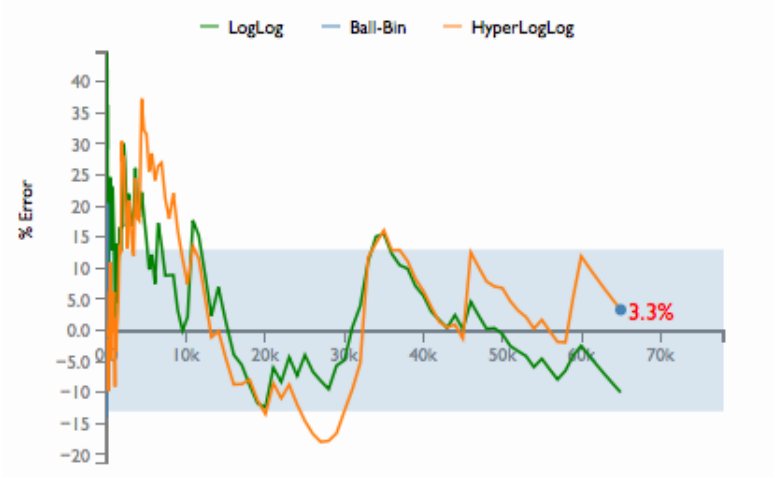
  - WRI$_OPTSTAT_SYNOPSIS$

$$rows = \sum_{col=1}^{number\ of\ columns} NDV(col)$$

trivadis makes IT easier.

What is HyperLogLog?

- Algorithm for approximation of number of distinct values (NDV)
- Bit-pattern observables on hash values of incoming data
- Published in 2007, extension of *LogLog* Algorithm



Source: http://content.research.neustar.biz/blog/hll.html

Further information:

- HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm (2007)
- Sketch of the Day: HyperLogLog – Cornerstone of a Big Data Infrastructure (2012)

trivadis makes IT easier.

The HyperLogLog Algorithm

Let $h : \mathcal{D} \to [0,1] \equiv \{0,1\}^\infty$ *hash data from domain* $\mathcal{D}$ *to the binary domain.*
Let $\rho(s)$, *for* $s \in \{0,1\}^\infty$, *be the position of the leftmost 1-bit* ($\rho(0001\cdots) = 4$).

Algorithm HYPERLOGLOG (**input** $\mathcal{M}$: multiset of items from domain $\mathcal{D}$).
assume $m = 2^b$ with $b \in \mathbb{Z}_{>0}$;
initialize a collection of m registers, $M[1], \ldots, M[m]$, to $-\infty$;

for $v \in \mathcal{M}$ **do**
  set $x := h(v)$;
  set $j = 1 + \langle x_1 x_2 \cdots x_b \rangle_2$; {*the binary address determined by the first b bits of* x}
  set $w := x_{b+1} x_{b+2} \cdots$; **set** $M[j] := \max(M[j], \rho(w))$;

compute $Z := \left(\sum_{j=1}^{m} 2^{-M[j]} \right)^{-1}$; {*the "indicator" function*}

return $E := \alpha_m m^2 Z$ with α_m as given by Equation (3).

Source: http://algo.inria.fr/flajolet/Publications/FlFuGaMe07.pdf

© 2007 Discrete Mathematics and Theoretical Computer Science (DMTCS), Nancy, France

trivadis makes IT easier.

Synopsis Calculation (≥ Oracle 12.2)

- Synopsis calculation with **HyperLogLog** (HLL) algorithm
- *Approximate* NDV stored in WRI$_OPTSTAT_SYNOPSIS_HEAD$
- Size of synopsis tables:
  - WRI$_OPTSTAT_SYNOPSIS_HEAD$

$$rows = number\ of\ partitions\ * number\ of\ columns$$

  - WRI$_OPTSTAT_SYNOPSIS$

$$rows = 0$$

Additonal information for HLL in column WRI$_OPTSTAT_SYNOPSIS_HEAD$.SPARE2

trivadis makes IT easier.

Configuration of Synopsis Calculation in Oracle 12.2

- Set Optimizer Statistics Preferences in DBMS_STATS

```
dbms_stats.set_database_prefs('APPROXIMATE_NDV_ALGORITHM', '<value>')
dbms_stats.set_global_prefs('APPROXIMATE_NDV_ALGORITHM', '<value>')
dbms_stats.set_schema_prefs('<schema>', 'APPROXIMATE_NDV_ALGORITHM', '<value>')
dbms_stats.set_table_prefs('<schema>','<table>','APPROXIMATE_NDV_ALGORITHM','<value>')
```

- Values for APPROXIMATE_NDV_ALGORITHM:

'REPEAT OR HYPERLOGLOG'	Preserve format for existing tables, create new tables with HyperLogLog (Default)
'ADAPTIVE SAMPLING'	Adaptive sampling is used for all synopses
'HYPERLOGLOG'	HyperLogLog is used for all new and stale synopses

trivadis makes IT easier.

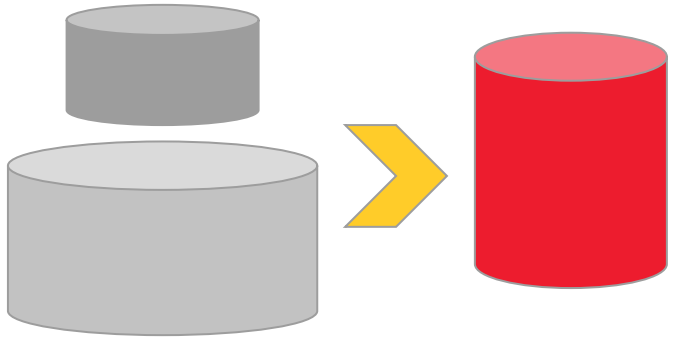
Migration Strategies

trivadis
makes IT easier.

Migration Strategies

- How to switch to the new synopsis calculation?

- There are different possible strategies
  - Don’t use HyperLogLog
  - HyperLogLog only for new tables
  - Replace all existing synopses



- See blog post of Nigel Bayliss:
  https://blogs.oracle.com/optimizer/efficient-statistics-maintenance-for-partitioned-tables-using-incremental-statistics-part-3

trivadis makes IT easier.

Migration Strategy “Conservative”

- ADAPTIVE SAMPLING
  - Leave it as it is (“I don’t trust this new stuff”)
  - No advantage of new synopsis calculation
  - Useful for backward compatibility
  - Not recommended as long term strategy!

```
dbms_stats.set_database_prefs
   ('APPROXIMATE_NDV_ALGORITHM', 'ADAPTIVE SAMPING');
dbms_stats.set_global_prefs
   ('APPROXIMATE_NDV_ALGORITHM', 'ADAPTIVE SAMPING');
```


trivadis makes IT easier.

Migration Strategy “Lazy”

- REPEAT OR HYPERLOGLOG
  - Only new tables will use HyperLogLog synopses
  - Existing tables continue with existing format
  - No migration effort required
  - No action needed, default setting

trivadis makes IT easier.

Migration Strategy “Modern”

- HYPERLOGLOG
  - Get rid of old-fashioned synopses
  - Highly recommended for tables with many partitions
  - If possible, use it for all partitioned tables
  - Needs to re-gather statistics on all partitioned tables

```
dbms_stats.set_database_prefs
   ('APPROXIMATE_NDV_ALGORITHM', 'HYPERLOGLOG');
dbms_stats.set_global_prefs
   ('APPROXIMATE_NDV_ALGORITHM', 'HYPERLOGLOG');
```


trivadis makes IT easier.

Conclusion

trivadis makes IT easier.

Conclusion

- Reduced space requirements in tablespace SYSAUX
  - WRI$_OPTSTAT_SYNOPSIS$ not needed anymore
  - WRI$_OPTSTAT_SYNOPSIS_HEAD$ grows (blob column SPARE2)
- Better performance for gathering incremental statistics
  - Problems of Oracle 12.1 seem to be solved
  - HyperLogLog algorithm is very efficient
- Different migration strategies
  - Good default behavior (no action required for most databases)
  - Flexible solutions for tables with many partitions

trivadis makes IT easier.

Thank you.

Dani Schnider, Trivadis AG

@dani_schnider danischnider.wordpress.com

BASEL ▪ BERN ▪ BRUGG ▪ DÜSSELDORF ▪ FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ GENEVA
HAMBURG ▪ COPENHAGEN ▪ LAUSANNE ▪ MUNICH ▪ STUTTGART ▪ VIENNA ▪ ZURICH

trivadis
makes IT easier.