SolidQ

Cardinality Estimator

Herbert Albert
Halbert@SolidQ.com

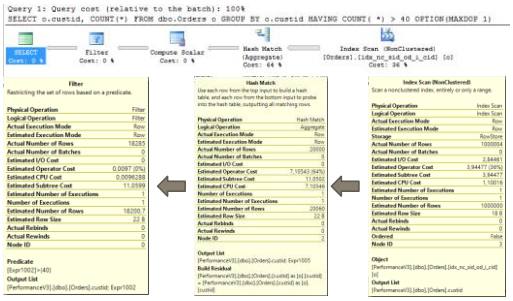
Agenda

- Cardinality Estimation Basics
- Statistics Basics
  - Definition
  - Creating and Updating Statistics
  - Best Practices
- Common Cardinality Estimation Problems
- The New Cardinality Estimator
  - Changes
  - Regression Problems and possible Solutions

Cardinality Estimation

- Cardinality Estimation is the process of estimating the
  - number of rows processed by the different operators in a query plan
  - distribution of values
  - distinct value counts
  - duplicate counts
- Essential for
  - Query plan generation
    - Finding the right indexes, joins, groups,…
  - Query execution
    - Memory Grants
    - DOP
- Largely based on SQL Server 7.0
  - Fixes in QFEs under trace flag to prevent regressions
  - Some problems needed a major redesign

Calculation per Operator



Agenda

- Cardinality Estimation Basics
- Statistics Basics
  - Definition
  - Creating and Updating Statistics
  - Best Practices
- Common Cardinality Estimation Problems
- The New Cardinality Estimator
  - Changes
  - Regression Problems and possible Solutions

Statistics

- Cardinality estimation is based on statistics
- In the absence of statistics or if statistics cannot be used for the estimation, SQL Server uses heuristics
  - Can led to horrible plans
- Statistics are created for one or a combination of columns
- Statistics can be created and updated automatically and manually
  - Each index has a statistic

Statistics

- Each statistic consists of
  - Header
    - Rows sampled, rows in table, last update timestamp,…
  - Density vector
    - Measure for the uniqueness of a column
    - Calculated by 1/(number of distinct values)
    - 1 x the number of rows returns the estimate for one value
    - Histogram is used when possible
  - Histogram
    - Data distribution of the column(s) with up to 200 steps
  - String summary
    - A statistic for the frequency distribution of substrings in a string column which is used for like predicates.

DBCC SHOW_STATISTICS

- General Information on statistics
- Histogram

AAADKHZRMP is the smallest value in the sample
There are about 61 rows with the value APQYHJBZRA
The range between ALPAAMGEBU and APQYHJBZRA has
- ~2213 rows
- 322 distinct values
- Average of 6,876072 row per values

	RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
1	AAADKHZRMP	0	1	0	1
2	ABWWWHCTG	899.0234	51.66787	123	7.293098
3	AGSXPCHXBW	2218.063	91.59305	399	5.553113
4	AJCPVATWG	1248.25	77.50181	171	7.296861
5	ALPAAMGEBU	1038.242	75.15327	186	5.583848
6	APQYHJBZRA	2213.344	61.06203	322	6.876072
7	ATJVVZKNLY	1649.389	108.0328	284	5.814801
8	AVYLATHSBP	1139.707	51.66787	204	5.586987

General Assumptions

- Independence
  - Distribution on different columns are independent except correlation information is available
- Uniformity
  - Within each histogram step, distinct values are evenly spread
- Containment and Inclusion
  - If something is searched for, it is assumed that it exists.
  - For example in equi-joins and filter predicates

Demo

- Statistic and Cardinality Estimation Basics

Creating Statistics

- Automatically
  - With every DML statement that needs statistics, if allowed to do so
    - Only single-columns statistics
    - No filtered statistics
- Explicitly
  - CREATE STATISTICS
  - CREATE INDEX
  - sp_createstats

Updating Statistics

- Automatically triggered through query compilation
  - Colmodctr exceeds threshold
  - Allowed to do so
    - Database level setting
    - Overruled by table level setting
    - Overruled by statistic level setting
    - Set through ALTER DATABASE, UPDATE STATISTICS and sp_autostats
- Manually
  - UPDATE STATISTICS
  - Index REBUILD
  - sp_updatestats

Update Statistics threshold

- Cardinality goes from 0 to 1
- Old statistics was created with less than 500 rows and colmodctr is greater than 500
- Old statistics with more than 500 rows and colmodctr > 500 + 20% of the rows
  - For filtered statistics the threshold is modified by the selectivity
- Temporary tables have an additional threshold with cardinality of 6

Managing Statistics

Auto Create Statistics

- SQL Server can create new statistics if needed by Query Optimizer

Auto Update Statistics

- SQL Server can update outdated statistics
- Based on a threshold on data changes
  - Tracked per column since SQL Server 2005

Auto Update Statistics Asynchronously

- Statistic is asynchronously update when query optimizer finds a outdated statistic during optimization
- Uses old statistic for the optimization that triggers the update

Manual Statistics

- CREATE / UPDATE STATISTICS
  - Ability to turn of automatic update
  - Can be used to create Multi-column statistics
  - Custom sample percentage
- sp_createstats, sp_updatestats

```
CREATE STATISTICS statistics_name ON {
table_or_indexed_view_name } ( column [ ,...n ] )
[ WHERE <filter_predicate> ]
[ WITH
[ [ FULLSCAN | SAMPLE number { PERCENT | ROWS } |
STATS_STREAM = stats_stream ] ]
[ [ , ] NORECOMPUTE ]
[ [ , ] INCREMENTAL = { ON | OFF } ]    ] ;
```

General Best Practices

- Keep Auto Options ON
  - Use opt-out for exceptions
    - Statistics that shouldn't be updated
    - Statistics that need a different sampling rate
- Update statistics regularly
  - Don't let the colmodctr trigger an update too often
  - 20% of changes might be too late

Demo

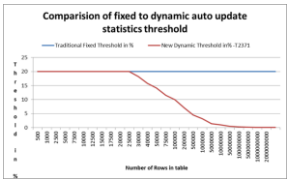
- Auto Options
- Querying Statistics

Agenda

- Cardinality Estimation Basics
- Statistics Basics
  - Definition
  - Creating and Updating Statistics
  - Best Practices
- Common Cardinality Estimation Challenges
- The New Cardinality Estimator
  - Changes
  - Regression Problems and possible Solutions

Challenges - Large Tables

- 20% might be too much as threshold for an update
- Trace Flag 2371
  - lower threshold based on # rows
  - https://support.microsoft.com/en-us/kb/2754171



Source: http://blogs.msdn.com/b/saponsqlserver/archive/2011/09/07/changes-to-automatic-update-statistics-in-sql-server-traceflag-2371.aspx

Challenges – Increasing Values

- Not an issue for highly selective queries
- Update more frequently
- Use Trace Flags for auto-quick-statistics
  - Trace Flag 2389
    - Enable update for known ascending keys
  - Trace Flag 2390
    - Enable also for keys marked as unknown
  - Trace Flag 4139
    - Enable also for keys marked as stationary
    - Since SQL 2012 SP1 CU10

Challenges - Multi-column filter

- Assume a query on a table with 10 Mio rows and 2 filters
  - Manufacturer = 'VW' estimated 5 Mio rows
  - Model = 'Golf' estimated 4 Mio rows
- How many rows should SQL Server estimate?
  - 4 Mio rows -> fully dependent
  - 0 rows -> excluding
  - 2 Mio rows -> independent (10x0,5x0,4)
  - Or anything else between 0 and 4 Mio rows?
- SQL Server considers the filters to be independent
- If that isn't the truth
  - Create a multi-column statistic
    - Multi columns statistics are not created automatically!
  - Filtered statistics might also be solution

Challenges - Heuristics

- Missing statistics
- Use of local variables in predicates
  - Recompile might be solution (be careful)
- Multistatement table-valued function
  - Use inline table-valued function when possible
- Table variables
  - No statistics
  - No idea of number of rows in the table
  - Use temporary tables
- Non-constant-foldable expressions
  - YEAR(orderdate) = 2015
  - Rewrite the expression
  - Use statistics on computed columns
- Comparison of columns
  - Shipdate > Duedate
  - Rewrite as expression and use computed column

Agenda

- Cardinality Estimation Basics
- Statistics Basics
  - Definition
  - Creating and Updating Statistics
  - Best Practices
- Common Cardinality Estimation Problems
- The New Cardinality Estimator
  - Changes
  - Regression Problems and possible Solutions

New Cardinality Estimator

- Old Cardinality Estimator
  - Largely based on SQL Server 7.0
  - Fixes in QFEs under trace flag to prevent regressions
  - Some problems needed a major redesign
- SQL Server 2014 CE
  - Statistics stays the same “only” the estimation changes
  - Can cause regressions
  - However most query will benefit from changes

New Cardinality Estimator

▶Configuration

- New Cardinality Estimator
  - Compatibility level 120
  - Trace flag 2312 (Query Level)
- Old Cardinality Estimator
  - Compatibility level 110
  - Trace flag 9481 (Query or System Level)
- Test for regression problems
- Extended Events
  - query_optimizer_estimate_cardinality
  - query_optimizer_force_both_cardinality_estimation_behaviors
- CardinalityEstimationModelVersion in Execution Plans

New Cardinality Estimator

▶Main changes

- Ascending data
  - 120 CE use average cardinality for each value in the column
- Same table filtered predicates
  - Now assumes there's some correlation
- Different tables filtered predicates
  - Now assumes more independence
- Joins
  - More stable algorithms
- Heuristics
  - Multi Statement Table-Valued Function assume 100 rows now

New Cardinality Estimator

▶Multiple filters on a single table

- Selectivity of a single filter is defined as
  - estimated divided by the total number of rows
- Old CE assumes no dependency with multiple filters
  - $Estimate = Sel1 \times Sel2 \times \ldots \times total\ rows$
- New CE estimates some dependency
  - Exponential backoff algorithm
  - $Estimate = Sel1 \times \sqrt{Sel2}\ x\ \sqrt[4]{Sel3}\ x\ \sqrt[8]{Sel4}$
  - Selectivity sorted from highest to lowest
  - Up to 4 filters used
- for highly dependent data also consider other solutions
  - multi-column statistics
  - Filtered statistics

Increasing Values

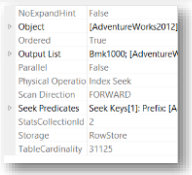
- Not an issue for highly selective queries
- The new estimator considers the same distribution as within the histogram
  - Trace Flags typically not needed anymore

New Cardinality Estimator

▶query_optimizer_estimate_cardinality

- Only for troubleshooting – debug channel

- Correlate with query plan using stats_collection_id



New Cardinality Estimator

▶Resources

- Miloš Radivojević Blog Series
  - http://milossql.wordpress.com/tag/cardinality-estimator
- A First Look at the New SQL Server Cardinality Estimator
  - http://www.sqlperformance.com/2013/12/t-sql-queries/a-first-look-at-the-new-sql-server-cardinality-estimator
- New functionality in SQL Server 2014 – Part 2 – New Cardinality Estimation
  - http://blogs.msdn.com/b/saponsqlserver/archive/2014/01/16/new-functionality-in-sql-server-2014-part-2-new-cardinality-estimation.aspx

