

SQL Server 2014 Cardinality Estimation

Bob Beauchemin
bobb@SQLskills.com



Outline

- Cardinality Defined
- Cardinality Estimator – SQL Server 7 – 2012
- Cardinality Estimator – SQL Server 2014
- Tracing and Troubleshooting the new Cardinality Estimator



2

© SQLskills. All rights reserved.
<http://www.SQLskills.com>



Definition

- **Cardinality in databases**
 - The relationship between two tables (e.g. one-to-many)
 - The uniqueness of data values contained in a column
- **In query processor**
 - How many rows will satisfy this iterator
 - Query plan optimization depends on cardinality estimator
- **Query processor uses**
 - Statistics (with all the limitations of statistics)
 - Constraints (e.g. unique constraint)
 - Heuristics (when parameter values are unknown)
- Estimates are based on a set of basic assumptions

Cardinality Usage

- **Estimates particularly important with**
 - Predicates – WHERE, HAVING, JOIN
 - Distinct Values – SELECT DISTINCT, GROUP BY
- **Affects plans in a number of ways**
 - Access method strategy
 - Join type and join order
 - Plan shape
 - Memory grants
 - Parallel vs. serial plan
 - Optimization efforts
- **Cardinality skews in earlier plan steps can affect later steps**

Basic Assumptions

- **Uniformity**
 - Distinct values are evenly spaced and have same frequency
- **Containment**
 - For column = constant, assume that constant exists for column
 - For equijoin, values of foreign keys exist
- **Independence**
 - No dependence between columns in same table
 - $\text{Sel}(P \wedge Q) = \text{Sel}(P) * \text{Sel}(Q)$
- **Inclusion**
 - Inside histogram boundaries, there is always is match
 - Number of matched rows = (histogram step cardinality * avg frequency)

Estimation Example

- **For a simple WHERE clause with single column statistics**
 - RangeHighKey - best
 - AVG_RANGE_ROWS - interpolate between range
 - Density of entire column (1/#distinct values)
 - Hardcoded guess (when parameterized and value is unknown)
- **Also known as "base statistics"**

Old Cardinality Estimator

- Introduced in SQL Server 7
- Extensions added over time
 - Through trace flags
 - Through version-compatibility features
- Over time it's hard to predict, understand, or control
- Customers hate plan regressions most of all

Cardinality Estimation Trace Flags

- **TF 2301 – Modeling Extensions**
 - Base Containment Assumption
 - Integer Modeling
 - Comprehensive Histogram Usage
 - Comprehensive Density Remapping
 - Comprehensive Density Matching
 - <http://blogs.msdn.com/b/ianjo/archive/2006/04/24/582219.aspx>
- **TF 2389/2390 - Handles ascending key problem**
 - Optimizer will brand a key ascending after a few statistics updates (TF 2389)
 - Statistics auto-updated at compile time
 - TF 2390 assumes every key column without branding
 - Adding TF 2388 displays ascending key info w/DBCC SHOW_STATISTICS
 - <http://blogs.msdn.com/b/ianjo/archive/2006/04/24/582227.aspx>

The Catch-All Trace Flag

- **TF 4199**
 - Incorporates all previous fixes made for query processor by trace flags
 - Not just cardinality estimates
 - Added in
 - SQL Server 2005 SP3 CU6
 - SQL Server 2008 and 2008 SP1 CU7
 - SQL Server 2008 R2 and SQL Server 2012
 - <http://support.microsoft.com/kb/974006/en-us> for list

Tracing Inaccurate Cardinality Estimation

- **An XEvent added in SQL Server 2012**
 - inaccurate_cardinality_estimate
- **Fires on some iterators**
 - That produce different number of rows than they consume
 - Can fire multiple times per iterator and per query
- **Calculates an inaccuracy threshold at execution time**
 - Base on number of rows estimated
- **Fires (and recalculates threshold) when the threshold is reached**
 - Actual number of rows reflects the number returned so far by that iterator

SQL Server 2014 Cardinality Estimator

- **Database compatibility level determines estimator**
 - Compatibility level 120 – new estimator
 - Older compatibility levels (70) – old estimator
- **Trace flags can control it**
 - TF 2312 – use new estimator regardless of compatibility level
 - TF 9481 – disable new estimator
- **For individual regressions, use query hint**
 - `OPTION QUERYTRACEON(2312)`
- **Query plans show which cardinality estimator used**
 - `CardinalityEstimationModelVersion = 70/120` – statement iterator properties

New Estimator Goals

- **Consistent, predictable query performance**
 - Different operator trees that represent the same relation have the same cardinality estimates.
- **New model for better performance**
 - Significant changes to the model result in more accurate cardinality estimates and better plan choices
- **Incorporate special case code**
- **Easier to support**
- **Minimal regression**

How it's different

- **SQL Server 7 cardinality estimator**
 - Integrated in query optimization
 - Special cases in code
 - Two wrong estimates sometimes made a right one (but not usually)
- **SQL Server 2014 estimator works in two stages**
 - Choose appropriate estimation calculator (~50 choices)
 - Execute the estimation process with the appropriate statistics
- **SQL Server 2014 estimator includes refinements of basic assumptions**

Refinements of Basic Assumptions

- **Independence vs. Correlation**
 - Old: Columns independent unless multi-column stat density says otherwise
 - New: Use exponential back off for multi-column, single table predicate
- **Containment – ascending key (new)**
 - New: Always assume query predicate value exists
 - New: Estimate cardinality using average frequency

Containment and Join Model Changes

- **Containment – multiple predicates in joined tables**
 - Old: Simple containment
 - All rows matching predicate in table A match predicate in table B
 - New: Base containment
 - All rows matching predicate in table A exist in table B (but don't always match)
- **Containment – simple join**
 - Old: Use histogram from each table
 - New: Assume keys in table A exist in table B
- **Overpopulated primary key enhancement**
 - Old: Use entire histogram range
 - New: If join column overpopulated, use only the part of range that correlates
 - Example: Fact table with overpopulated date dimension table that includes future dates

Model Variations

- **Trace flags permit going back to old model**
 - Used for regressions and individual model changes
- **TF 9476 - Use simple containment instead of base containment**
- **TF 9481 - Ignore overpopulated primary key enhancement**
- **TF 9489 - Do not use ascending key enhancement**

Standard trace flag disclaimers apply...

New DMV Provides Estimate vs. Actual

- **sys.dm_exec_query_profiles** provides information about queries during query execution
- **Counts reported are**
 - Per-operator
 - Per-thread
- **You can compare estimated to actual rows while query is running**
 - Requires "show actual query plan" is on
- **Additional information**
 - Can be joined with performance counters

Trace Flags for Cardinality Estimate Info

- **Trace flag 8612 can outputs some cardinality info**
 - SQL Server 2014 and previous releases
- **Trace flag 2363 outputs additional selectivity info**
 - SQL Server 2014 only
- **Usually used with trace flags that show optimization process**

Tracing Cardinality Estimation

- **A single XEvent produces detailed output**
 - query_optimizer_estimate_cardinality
- **One query produces multiple cardinality events**
 - Number of events proportional to complexity of statement
 - Optimizer estimates cardinality after query transformations as well
- **Each event produces three major sets of output**
 - InputRelation – Logical operator and part of a parse tree
 - Use to correlate to query plan operators
 - Some operators contain cardinality number
 - Calculator – Info about cardinality calculator types and calculators used
 - StatsCollection – Statistics read for this estimation step and estimates
- **Note: Old cardinality estimator does not emit this event**

Cardinality Calculator Types (Undocumented)

- DistinctCountCalculator
- FilterCalculator
- JoinCalculator
- ResidualCalculator
- SpecialCalculator

Cardinality Calculators (Undocumented)

- | | |
|---|---|
| <ul style="list-style-type: none"> ▪ CDVCPanCollapse ▪ CDVCPanFilter ▪ CDVCPanJoin ▪ CDVCPanLeaf ▪ CDVCPanProjectOrGroupBy ▪ CDVCPanRedundantOJ ▪ CDVCPanRemap ▪ CDVCPanSplit ▪ CDVCPanTrivial ▪ CDVCPanUnionAll ▪ CDVCPanUniqueKey ▪ CDVCPanUpdate ▪ CSeCalcAscendingKeyFilter ▪ CSeCalcAscendingKeyJoin ▪ CSeCalcBitmapFilter ▪ CSeCalcColInIntvlWithFilteredStat ▪ CSeCalcColumnInInterval ▪ CSeCalcCombineFilters_ExponentialBackoff ▪ CSeCalcCombineFilters_Independence ▪ CSeCalcCombineFilters_MinSelectivity ▪ CSeCalcExpressionComparedToExpression | <ul style="list-style-type: none"> ▪ CSeCalcFilteredStatsWrapper ▪ CSeCalcFilterWithResidualGuess ▪ CSeCalcFixedFilter ▪ CSeCalcFixedJoin ▪ CSeCalcFunctionComparisons ▪ CSeCalcGuessComparisonJoin ▪ CSeCalcGuessInList ▪ CSeCalcHistogramComparison ▪ CSeCalcIndependentJoin ▪ CSeCalcMultiColumnEqualsMultiColumn ▪ CSeCalcMultiColumnSemiJoin ▪ CSeCalcNegativeFilter ▪ CSeCalcNegativeJoin ▪ CSeCalcOneSided ▪ CSeCalcPointPredsFreqBased ▪ CSeCalcPointPredWithFilteredStat ▪ CSeCalcPureHiddenSide ▪ CSeCalcSimpleJoin ▪ CSeCalcSimpleJoinWithDistinctCounts ▪ CSeCalcTrieBased ▪ CSeCalcTwoStepConjunction ▪ CSeCalcUniqueKeyFilter |
|---|---|

CSeCalc = Class Selectivity Calculator

CDVC = Class Distinct Value Calculator



21

© SQLskills, All rights reserved.
<http://www.sqlskills.com>

Cardinality Statistics Collections(Undocumented)

- | | |
|---|---|
| <ul style="list-style-type: none"> ▪ CStCollBaseTable ▪ CStCollBlackBox ▪ CStCollCollapse ▪ CStCollConstTable ▪ CStCollFilter ▪ CStCollFilteredStatsShim ▪ CStCollFudge ▪ CStCollGroupBy ▪ CStCollIntervalFilterPlanned ▪ CStCollJoin ▪ CStCollJoinExt | <ul style="list-style-type: none"> ▪ CStCollOrderedTop ▪ CStCollOuterJoin ▪ CStCollProject ▪ CStCollRemap ▪ CStCollRemoteTable ▪ CStCollSingleColumnAntiSemiJoin ▪ CStCollSplit ▪ CStCollSTVF ▪ CStCollUnionAll ▪ CStCollUpdate |
|---|---|



22

© SQLskills, All rights reserved.
<http://www.sqlskills.com>

Cardinality Estimates and Query Plan

- **When query_optimizer_estimate_cardinality is on**
 - Each trace record contains a stats_collection_id property
- **Query plans iterators contain StatsCollectionId property**
- **These can be correlated**
 - Which trace records refer to each iterator
 - Not every iterator has corresponding trace records

Summary

- **Accurate cardinality estimation critical to good query plan**
- **New Cardinality Estimator introduced in SQL Server 2014**
 - Chooses calculator to use
 - Uses calculator and statistics to calculate cardinality
- **Refinements to basic assumptions**
- **You can choose new or old estimator**

References

- **Optimizing Your Query Plans with the SQL Server 2014 Cardinality Estimator, Joe Sack**
 - <http://msdn.microsoft.com/en-us/library/dn673537.aspx>
- **Testing Cardinality Estimation Models in SQL Server**
 - <http://dl.acm.org/citation.cfm?id=2304526>
- **Ian Jose's Blog**
 - <http://blogs.msdn.com/b/ianjo/>
- **My Blog**
 - <http://www.sqlskills.com/blogs/bobb/>
- **Joe Sack's Blog**
 - <http://www.sqlskills.com/blogs/joe/>
- **Dima Pilugin's Blog**
 - <http://www.queryprocessor.com/>
- **A First Look at the New SQL Server Cardinality Estimator, Ben Navarez (On SQLPerformance.com)**