SQLSaturday 2017

Sioux Falls, SD | Hosted by (605) SQL

- Please be sure to visit the sponsors during breaks and enter their end-of-day raffles!
- Remember to complete session surveys! You will be emailed a link after the event or you visit http://www.sqlsaturday.com/662/Sessions/SessionEvaluation.aspx
- Event After Party
  - At Will's Training Table (Formally Beef O'Bradys) near the Pentagon starting at 5:45 PM
- Want More Networking and Training?
  - (605) SQL meets the 2nd Tuesday of every month. https://605sqlusergroup.sqlpass.org

Susantha Bathige

PASS SQLSATURDAY

Statistics And New Cardinality Estimator (CE)

Section Title

About me

- MCSA Database Administrator (SQL Server 2012/2014)
- MCITP SQL Server 2005
- Sr. DBA Specialized in SQL Server
- Working as a Production DBA at Pearson
- Has more than 12+ years of experience in SQL Server starting with SQL Server 7.0
- Writer: MSSQLTips.com, Personal blog (www.sqlservertorque.net)
- Active member in SQL Server User Groups in Denver and Sri Lanka
- Working on other database systems such as NoSQL DB systems like Cassandra and MongoDB
- Twitter: @sbathige Email: bathige@hotmail.com

In this section...

Part 1 – Introduction

Part 2 – Statistics

Part 3 – New CE model

Part 4 – Statistics Maintenance

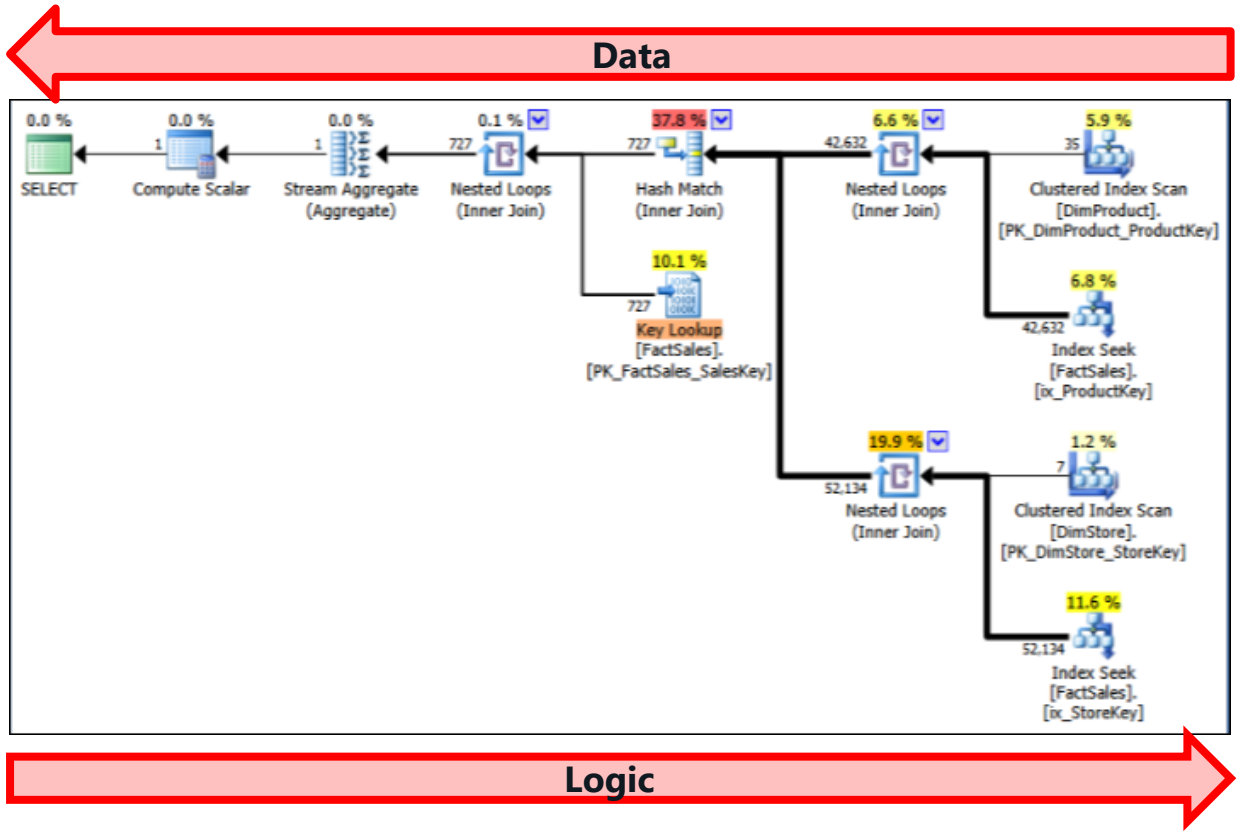
Part 5 - Recap

Introduction

Issues with execution plans

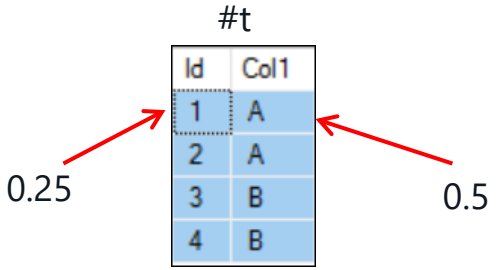
- Query timeout
- Inconsistent performance of stored procedures
- High tempdb utilization
- High CPU utilization
- Over utilized memory

Execution plan



Some concepts

- Predicate
  - Condition which can evaluate to TRUE, FALSE or UNKNOWN.
- Density
  - Uniqueness of a column.
  - Density = 1 / No. of distinct values
  - Ranges from 0 to 1.



- Estimates/Cardinality estimation
  - Estimated no.of records return by predicates (WHERE, HAVING, JOIN) or GROUP BY operations.
- Selectivity
  - Concept similar to cardinality estimation.
  - Percentage of rows that satisfy by a predicate.
  - Highly selective predicate returns small no.of records.

Understanding density

Low Density

High Selectivity

Returns less no.of records

More unique values

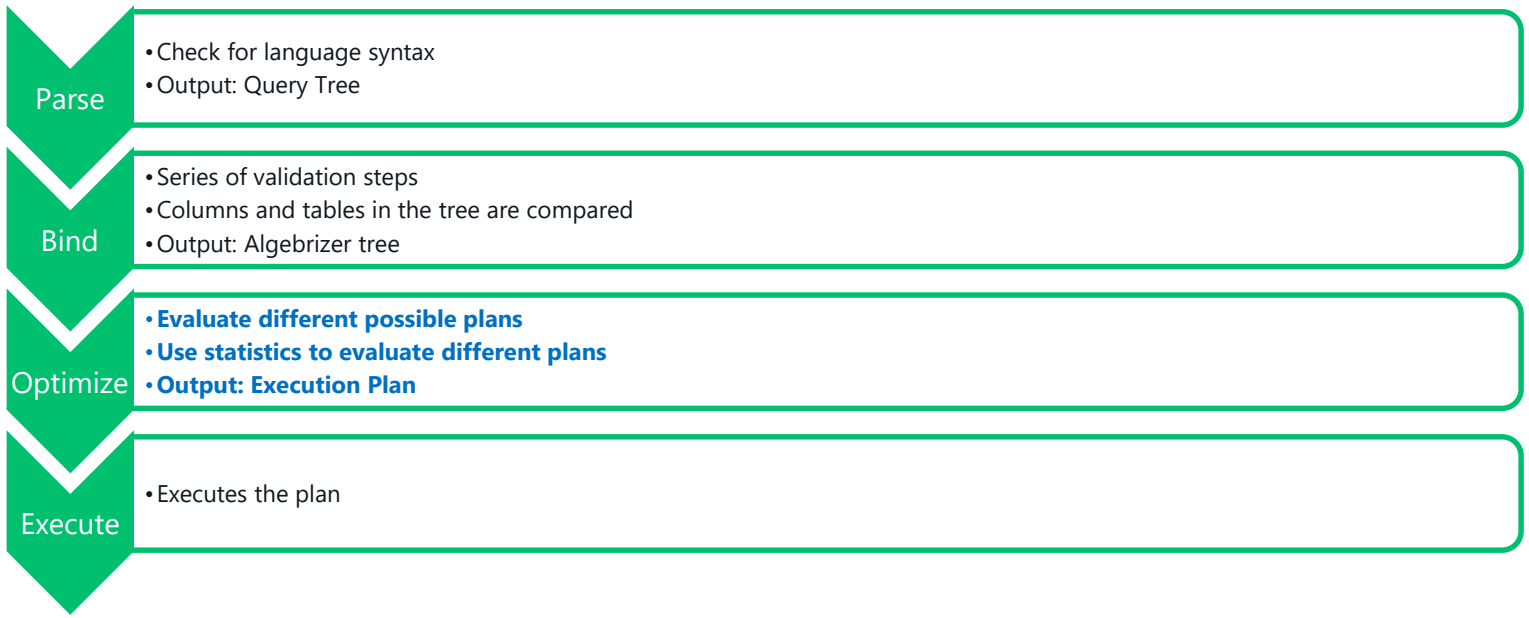
High Density

Low Selectivity

Returns high no.of records

Less unique values

How SQL Server generates Exec. Plan



Plan types

Optimal

- Fast execution
- Less resource consumption

Non optimal

- Slow execution
- High resource consumption

Statistics

Statistics

- Type of object exists in the database.
- Uses to generate execution plan.
- Can be associated with and index or they can exists with table column.
- Can create manually or query optimizer will create it for you.
- Has three main pieces;
  - Header
  - Density Vector
  - Histogram

Inside stats object

	Name	Updated	Rows	Rows Sampled	Steps	Density	Average key length	String Index	Filter Expression	Unfiltered Rows
1	IX_SalesOrderDetail_ProductID	Mar 14 2012 1:14PM	121317	121317	200	0.0078125	12	NO	NULL	121317

Header

	All density	Average Length	Columns
1	0.003759399	4	ProductID
2	8.242868E-06	8	ProductID, SalesOrderID
3	8.242868E-06	12	ProductID, SalesOrderID, SalesOrderDetailID

Density Vector

	RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
19	730	0	288	0	1
20	732	0	130	0	1
21	738	154	600	2	77
22	741	167	94	1	167
23	742	0	288	0	1
24	743	0	481	0	1
25	747	89	289	2	44.5

Histogram

Estimates vs actuals

- What is actuals? Figures after the query execution.
- What is estimates? Pre-determined values.
- Why SQL Server uses estimates? To create best possible Execution plan.
- When it uses estimates? At query optimization stage.
- How SQL Server uses estimates? We will go through in detail.

Cardinality estimator

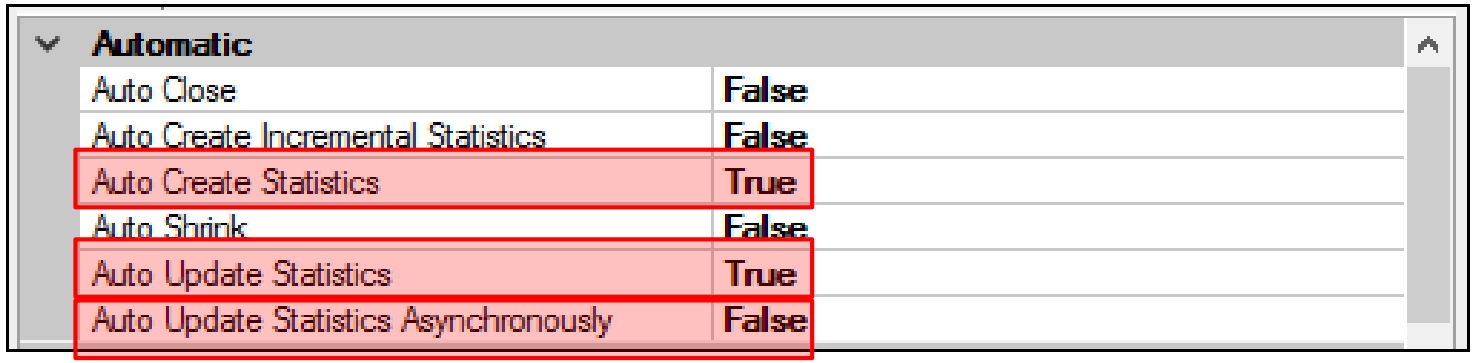
Calculate estimated no. of row counts for each operator within a query Exec. Plan

CE and estimates

- How many rows will satisfy a single filter predicate? Or multiple filter predicates?
- How many rows will satisfy a join predicate between two tables?
- How many distinct values do we expect from a specific column? A set of columns?
- Major factor in deciding which physical operator and plan shapes

DEMO – 1: Calculate Estimated Rows

DB options for stats



Importance of accurate cardinality estimates

- Memory
- Access Method
  - Table Scan
  - Index Scan
  - Clustered Index Scan
  - Index Seek
  - Clustered Index Seek
  - RID Lookup
- Join Type (Nested Loop, Merge, Hash)

New CE model

New CE model

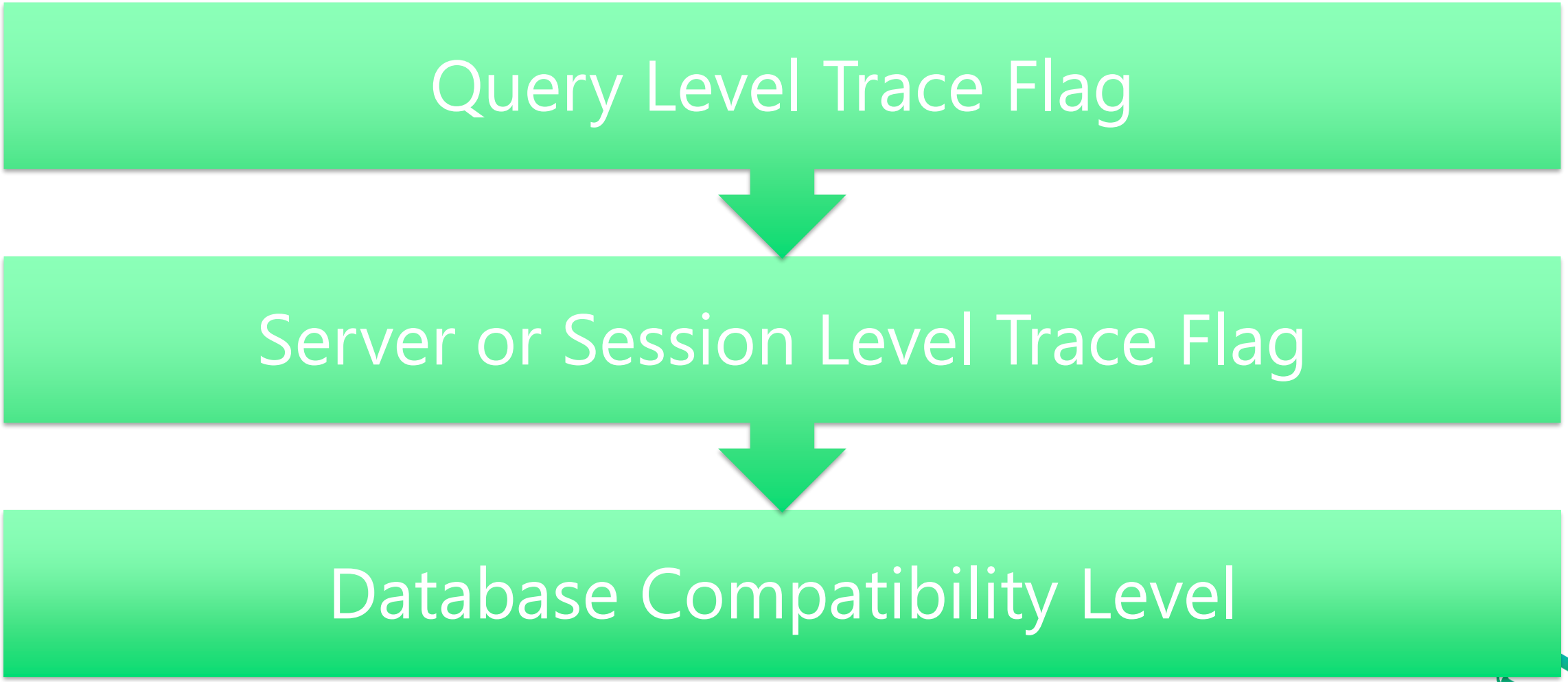
- Introduced in SQL Server 2014 (C.L 120)
- Major redesign of CE model after SQL Server 7.0 (C.L 70)
- There are some changes in SQL Server 2016 too.
- Affect Exec. Plan quality

While many workloads will benefit from the new CE, in some cases, workload performance may degrade without a specific tuning effort.

How to activate new CE model

- ALTER DATABASE [AdventureWorks2016] SET COMPATIBILITY_LEVEL = 120;
- TF 2312
  - Server level - DBCC TRACEON(2312,-1)
  - Session level - DBCC TRACEON(2312)
  - Query level - QUERYTRACEON 2312
- To revert – TF 9481

CE precedence



DEMO – 2: The impact of new CE model

Under and over estimates of row counts

Under estimates

- Selection of serial plan when parallelism would have been an optimal method.
- Inappropriate join strategies.
- Inefficient index selection and navigation strategies.

Over estimates

- Selection of parallel plan when serial plan would be an optimal method.
- Inappropriate join strategies.
- Inefficient index navigation strategies.
- Inflated memory grants.

How do you know which CE model is using

- In XML plan
- XE
- Plan property window

Stats maintenance

- How to create stats

- Automatically by the Query Optimizer
  - AUTO_CREATE_STATISTICS DB option
  - Always single column statistics
- Explicit creation by using CREATE STATISTICS statement
  - Single or multi column statistics
- When an index is created
  - Single or multi column statistics

How to update stats

- AUTO_UPDATE_STATISTICS ON (DB option)
  - Automatically updates when they are out of date.
  - Synchronous (default).
  - Happens before optimization.
- AUTO_UPDATE_STATISTICS OFF
  - Asynchronous – happens after optimization.
- Index rebuild operation
- UPDATE STATISTICS statement

Note: Index Reorganization does not update statistics not even index statistics.

Stats maintenance

- UPDATE STATISTICS dbo.SalesOrderDetail
- UPDATE STATISTICS dbo.SalesOrderDetail WITH SAMPLE 50 PERCENT
- UPDATE STATISTICS dbo.SalesOrderDetail WITH FULLSCAN, COLUMNS
- UPDATE STATISTICS dbo.SalesOrderDetail WITH FULLSCAN, INDEX
- UPDATE STATISTICS dbo.SalesOrderDetail WITH FULLSCAN
- UPDATE STATISTICS dbo.SalesOrderDetail WITH FULLSCAN, ALL
- ALTER INDEX ix_ProductID ON dbo.SalesOrderDetail REBUILD

New feature in SQL Server 2016 SP1 CU4

Persisting statistics sampling rate

```
UPDATE STATISTICS [Sales].[SalesOrderHeaderBulk] [IX_OrderDate] WITH FULLSCAN,
PERSIST_SAMPLE_PERCENT = ON
```

Recap

Recap

- Up to date statistics are the main key factor for quality execution plan.
- Statistics needs be maintained.
- Multi-column statistics can be used to improve query performance when there is correlation between columns.
- Filtered statistics can be used to improve query performance in certain situations. Create many if needed.
- While many workloads will benefit from the new Cardinality Estimator changes, in some cases, workload performance may degrade without a specific tuning effort.
- Use parameters instead of variables in stored procedures.

Recap…

- Auto Create Statistics should be ON.
- Consider asynchronous statistics update only if your facing delays in query execution.
- Explicitly created statistics should be maintained otherwise they will easily get outdated.

References

SQL Server Technical Article:

Optimizing Your Query Plans with the SQL Server 2014 Cardinality Estimator

Research paper:

A Black Box Approach to Query Cardinality Estimation

Testing Cardinality Estimation Models in SQL Server

https://blogs.msdn.microsoft.com/psssql/2014/04/01/sql-server-2014s-new-cardinality-estimator-part-1/

THANK YOU SPONSORS

Thank You For Attending !