DESIGNING FOR PERFORMANCE SERIES

Smokin' Fast Queries – Query Optimization

Jennifer Smith, MCSE

jambam Technology Solutions

Agenda

- Statistics
- Execution plans
- Cached plans/recompilation
- Indexing
- Query writing tips
- New performance features in SQL

Statistics and the Query Optimizer

- Statistics are created for any index and can be created on non-indexed columns
  - Statistics are automatically created/updated with a FULL SCAN of the table for any index created/rebuilt
  - If a column has no index, SQL Server will create statistics for a column (based on sample size and only if AUTO CREATE is enabled)
- Statistics are made up of Density and Distribution of data for a given column
- Statistic information is stored in a histogram

```
Name                           Updated                 Rows                    Rows Sampled            Steps                   Den
------------------------------ ----------------------- ----------------------- ----------------------- ----------------------- ---
IX_Orders_FirstName            Jun 11 2016 10:57AM     223753774               744688                  74                      0

All Density                    Average Length          Columns
------------------------------ ----------------------- ------------------------------------------------------------------------
0.01351351                     5.056037                FirstName
1.342844E-06                   9.056037                FirstName, OrderID

Histogram Steps
RANGE_HI_KEY                   RANGE_ROWS              EQ_ROWS                 DISTINCT_RANGE_ROWS     AVG_RANGE_ROWS
------------------------------ ----------------------- ----------------------- ----------------------- -----------------------
Aaron                          0                       139716.9                0                       1
Alberta                        0                       139716.9                0                       1
Allyson                        0                       139116                  0                       1
```

DID YOU KNOW…?
A big difference between a Temporary Table and a Table Variable is a Table Variable has no statistics.

jambam Technology Solutions

Statistics – Best Practices

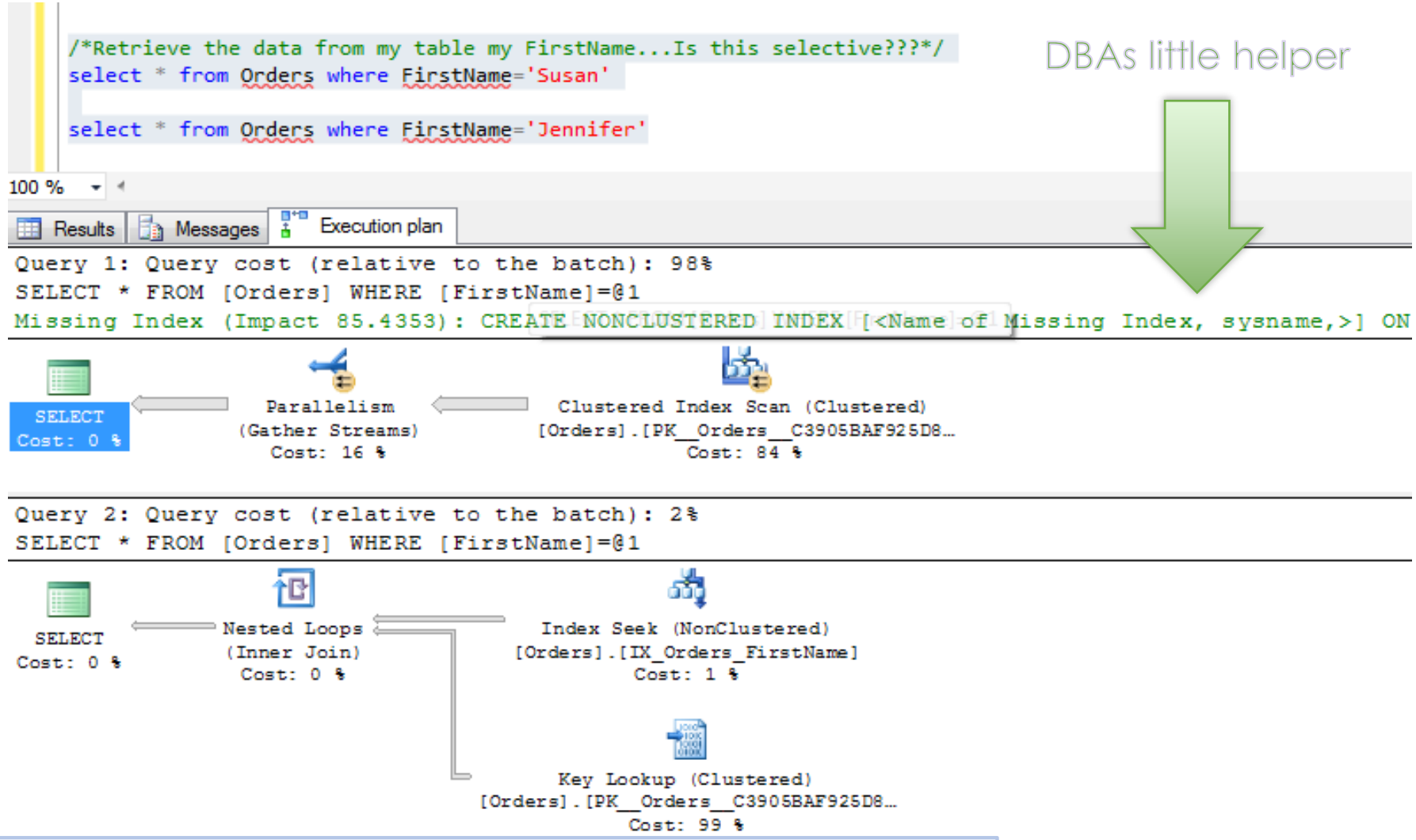
- Auto Create/Auto Update Statistics ON!!!!!!!!!! **unless turning off is a vendor requirement
- DO NOT rely on Auto Update
- Monitor your server for cases where estimated and actual statistics are significantly off (usp_CheckStatistics or 3rd party tool)
- Know your data distribution (and adjust sample rate accordingly).
- DO NOT update statistics after Creating or Rebuilding an Index

DID YOU KNOW…?
No Statistics will be used when retrieving data through a linked server.

jambam Technology Solutions

Execution Plans

- Things to look for (not always bad):
  - Table Scans
  - Key/RID Lookups
  - Join types – Primarily HASH joins
  - Sorts
  - Costs



SELECTIVITY IS VERY IMPORTANT WHEN IT COMES TO INDEX SELECTION

jambam Technology Solutions

Execution Plans

DEMO

To Plan Cache or Not to Plan Cache

- SQL Server has a pool of memory that is used to store both execution plans and data buffers.
- Query plans are “Good Enough” and time to build is variable based on query complexity
- Stored Procedures by default will cache plans
  - Good when data relatively evenly distributed and or query is extremely complex
  - Bad for Dynamic SQL and when results DRASTICALLY differ given the provide parameter
- Ad-Hoc plans are stored, note that typically since they are not “SAFE”…they are not reused
- Prepared (Parameterized) SQL

```
DECLARE @SqlCommand nvarchar(500)
  = N'SELECT count(*)
      FROM dbo.Orders AS p
      where OrderID=@OrderID';
DECLARE @ParameterDefinition nvarchar(100) = N'@OrderID int';
EXEC sp_executesql @SqlCommand, @ParameterDefinition,@OrderID=9660629;
```

jambam Technology Solutions

Have you heard???

- My database is slow…but I reboot it and everything works again… “for a while”!
- When I run the stored procedure for the current month, the query runs great…but year to date is horribly slow!
- My .NET developer added an index hint to that stored procedure last month…and it was soooo fast…but now it is slow again…I just don’t get it!
- This stored procedure has a bunch of optional parameters…but they are all pretty selective…I don’t get why it is so slow!

YOU PROBABLY HAVE A BAD QUERY PLAN!!!

jambam Technology Solutions

Plan Cache

DEMO

Indexing Best Practices

IT DEPENDS…but…generally…

- Columns being filtered or joined on are generally part of the main index column list and columns being selected should be in the include list
- As a general rule…adding an includes column set is never a bad thing.
- Consider reusability when creating indexes.
- A Non clustered index is most useful when it is at least 95 percent selective
- The most selective column in the index should be the first column…but only if it is frequently used as a filter…
- DO NOT create every index SQL Server recommends…especially in highly transactional systems

Top Performance Related Querying Mistakes (and how to fix them)

DEMO

jambam Technology Solutions

Query Writing Best Practices

- Watch out for Implicit Conversions
- Avoid “select *” where possible, but if all data is needed, there are scenarios that application developers will need it
- Make your queries selective where possible
- Use WITH (NOLOCK) where dirty reads are acceptable
- Consider which objects should use cached query plans and whether they will help your query perform better.
- Avoid performing calculations or conditions on table columns in the where clause that are being used as filters

DID YOU KNOW…?
The primary key of a table is stored at the node level of every index!

jambam Technology Solutions

New Performance Impacting Features in 2012/2014

- Newsequentialid()
- Optimize for ad hoc workloads
- OPTION (OPTIMIZE FOR (@?=?))
- In-Memory OLTP (Hekaton)
- Resource Governor for IO
- Data Files in Azure
- Delayed Durability

Summary

- Auto stats should be left on and know your data!
- Check your execution plans…if you add a column to a query adjust indexes accordingly
- Know when to recompile and parameterize
- Help SQL Server by writing efficient queries
- Always be on the lookout for enhanced features that allow you to perform operations more effectively

COMMENTS OR QUESTIONS?

Email - jennifer@jambam.com

Twitter - @dataninja_js