VISCOSITY

PROFESSIONAL SERVICES

VISCOSITY
PROFESSIONAL SERVICES

SQL Server Advanced SQL Server Performance Tuning

From Execution Plans to Extended Events

Download slides and code at

www.stormrage.com/

Lowell Izaguirre

Senior Principal Consultant

Viscosity North America

@viscosityna

linkedin.com/in/lowellizaguirre/

Lowell.Izaguirre@viscosityna.com

Agenda

- Dive deep into real-world tuning techniques, how to analyze query performance, and advanced Extended Events usage. You'll learn about high level procedure performance looking at the DMV,s and what my specific queries might help you identify as pain points.

- We will cover how to use extended events to capture things like specific procedure parameters, as well as a few other techniques for capturing measure related to performance or deadlocks.

- Then, we'll start with the top things I look for in a static code review, before I ever look at the execution plan. Those spot checks often find things that make a working procedure work substantially faster.

- Next, we will go over some real world execution plan examples, what they show us, and how to fix them. You'll see the basics like seek vs scan, sure, but what do things like key lookups, lazy table spool operators, and other things mean for you to review.

- We will finish with the code I use every day for finding missing indexes, which is very dense data wise, but once you start using it, you'll be a DBA Superhero.

Very Few Slides, Heavy Demo and Examples

- Wide open for questions as we go.
- I get nervous, and tend to go too fast. Help me slow down.
- Don’t panic how quick my Snippets allow me to get code instantly.

Procedure performance looking at the DMV's

- By Average Time
- By Last Elapsed Time
- By Max Logical Reads
- By Max Spills to TempDB

- Review the trivial differences in code
- Review the valuable columns in the results
- Review some execution plans from the results

Extended Event Magic Wand

- Reads the Columns of the Extended Event you created
- Creates a function to read all the columns
- Creates a permanent table to hold the results
- Creates a procedure which inserts into the table by reading the function
- Creates a procedure to toggle the ring buffer off and on to avoid reading the same events twice
- Creates a job to call the procedure every x minutes
- Creates a Report to send a recap every Friday
- Three of my must have Extended Events included.

Missing Indexes

- Missing index overview
- Review Details Of suggestion, first four examples.

Static Code Review

- Loops through your current sys.sql_modules for various points to investigate
- Various pain points
- implicit conversions scan
- Nested / multiple cte code review
- Objects that reference scalar operators

Standard Code Review Line Items

1. Some operations force single threaded operations instead of using maxdop: scalar UDF, multi-statement table-valued UDF,TOP, CLR,row_number or other windowed functions,system tables, object_name functions, referencing a view that has a scalar functions
2. Joins are Sarg-able, meaning all data types exactly match on both sides of the equals, no functions on columns, no implicit conversions
3. WHERE is Sarg-able, meaning all data types exactly match on both sides of the equals, no functions on columns, no implicit conversions
4. @Parameter data types and sizes exactly match the size of columns they are used against
5. Replace @TableVariables with #Temp tables to leverage statistics, unless the @TableVariable has less than 100 rows in every situation.
6. multiple CTEs or cascading CTEs should be replaced with #Temp tables to divide and conquer for better performance
7. anything that uses a VIEW or multiple views needs to be replaced with the underlying tables instead to eliminate duplicate or extra tables, remove convenience for performance.
8. convert any scalar functions to inline table value functions, and changes to use OUTER APPLY/CROSS APPLY for values.
9. Because FOR XML concatenation is performed early, any concatenation statements must come from a #temp table and not the source, as the WHERE statements are applied after FOR XML is performed.
10. uses SET NOCOUNT ON in first lines of the procedure
11. uses SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED in first lines of the procedure as appropriate --remove WITH(NOLOCK) hints in favor of this isolation level
12. avoid INSERT INTO #temp WITH ORDER BY(Adds unneeded sort operation)
13. avoid EXISTS featuring TOP (WHERE EXISTS(SELECT TOP 1 1.....
14. avoid UPDATE on OriginalTableName instead of Alias to avoid a implicit accidental cross join of original table
15. parallelism breakers: scalar and MSTVFs,TOP,rCTE,anything in sys schema or functions,CLR,Sequence,Aggregates,row_number()

Better Coding Practices:

1. Relevant recent comment on every deployed procedure or view
2. Every table should be fully qualified as SchemaName.TableName
3. Every table should have an alias with Table AS AliasName
4. Every column should have alias.columnname
5. Handles temp tables proactively, and in both try and catch due to connection pooling issues from java drivers
6. Try-Catch is being used as a best practice
7. Procedure Parameters example in the comments is populated with relevant data for better dev testing and error reporting
8. No nvarchars if not relevant
9. avoid SELECT *(Enumerate columns, avoid unneeded)
10. parallelism breakers: scalar and MSTVFs,TOP,rCTE,anything in sys schema or functions,CLR,Sequence,Aggregates,row_number()

Tiny Fraction of our SQL Server Health Check

VISCOSITY

PROFESSIONAL SERVICES

Thank You!

VISCOSITY

PROFESSIONAL SERVICES

Lowell.Izaguirre@ViscosityNA.com