Diagnosing Activity-related Database Issues

Glenn Berry
PRINCIPAL CONSULTANT – SQLSKILLS.COM

@GlennAlanBerry www.sqlskills.com/blogs/glenn

Module Overview

Diagnosing activity-related database issues

Interpreting query results

Alleviating activity-related database issues

Lock Waits

Tables and indexes that have lock waits

Very useful if there are very high average task counts

Shows row lock waits

Shows page lock waits

Cumulative waits since last SQL Server restart

Index tuning can often reduce lock waits

Lock Waits

Look for tables that have high row and/or page lock waits

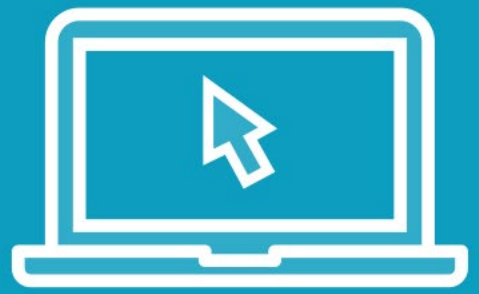
Look for example of high lock waits on clustered index of a table

Adding useful non-clustered indexes can help reduce lock waits

Dropping unused indexes can help reduce lock waits

Isolation level properties have a role in concurrency/locking/blocking issues

Demo



Lock Waits

Scalar UDF Statistics

Scalar UDF metrics for current database

Results are ordered by total worker time

Helps find most CPU-intensive scalar UDFs

Scalar UDF Statistics

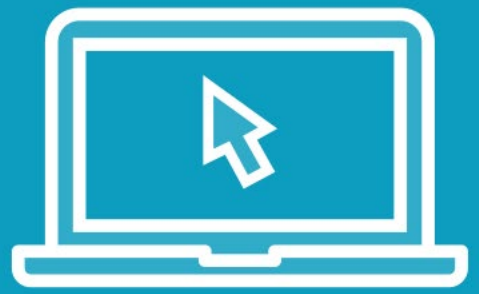
Scalar UDFs have known performance issues in SQL Server 2017 and earlier

Consider in-lining scalar UDF code if possible

Can convert to table-valued UDF that returns one column and row

Alternative is to convert scalar UDF to a T-SQL stored procedure

Demo



Scalar UDF Statistics

Input Buffer

Replacement for DBCC INPUTBUFFER

Shows last query for each SPID connected to current database

Returns useful performance metrics for each SPID

Input Buffer

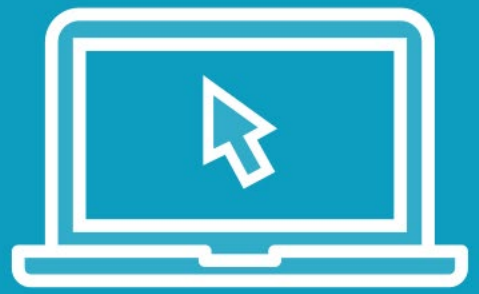
More capable and flexible replacement for DBCC INPUTBUFFER

Use for getting a quick overview of current query workload

Can add ORDER BY clause to focus on one specific area

Helps identify resources of long running queries that are still executing

Demo



Input Buffer

Query Execution Counts

Most frequently executed queries for current database

Look for “Has Missing Index” column

Look at graphical execution plan

Query Execution Counts

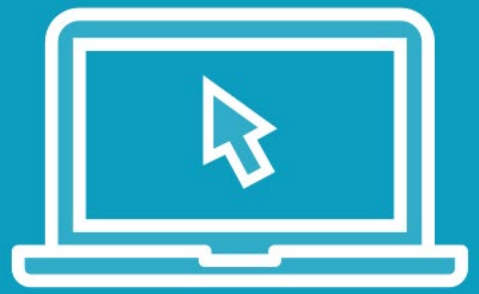
Helps understand baseline query workload

Frequently executed queries may be candidates for middle-tier or client-side caching

Extremely high counts may indicate application logic issues

Helps identify possible query and index tuning opportunities

Demo



Query Execution Counts

SP Execution Counts

Most frequently executed SPs for current database

Look for “Has Missing Index” column

Look at graphical execution plan

SP Execution Counts

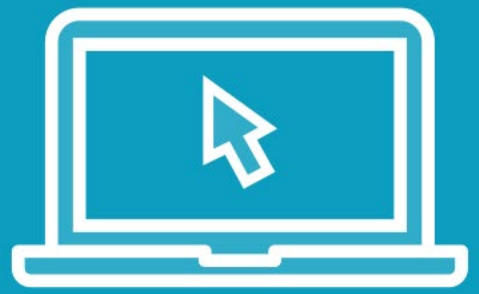
Helps you understand your baseline stored procedure workload

Frequently executed SPs may be candidates for middle-tier or client-side caching

Extremely high SP counts may indicate application logic issues

Helps you identify possible SP and index tuning opportunities

Demo



SP Execution Counts

SP Avg Elapsed Time

Cached stored procedures ordered by average execution time

Elapsed times are in microseconds

Look for large differences between min and max

Look for “Has Missing Index” column

Look at the graphical execution plan

SP Avg Elapsed Time

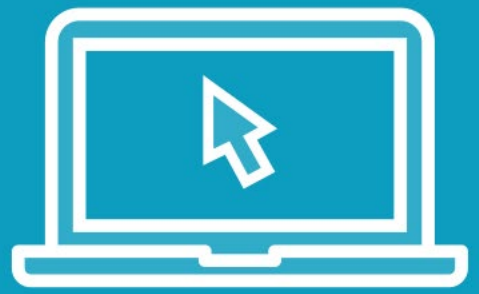
Helps identify possible easy tuning opportunities

Wide variance in execution times can indicate plan stability problems

Focus your initial tuning efforts on top five results

Dramatically reducing elapsed time of a stored procedure is very beneficial!

Demo



SP Avg Elapsed Time

Bad Nonclustered (NC) Indexes

Returns NC indexes that have more writes than reads

Consider dropping these indexes after more analysis

Make sure you know how long instance has been running

Bad NC Indexes

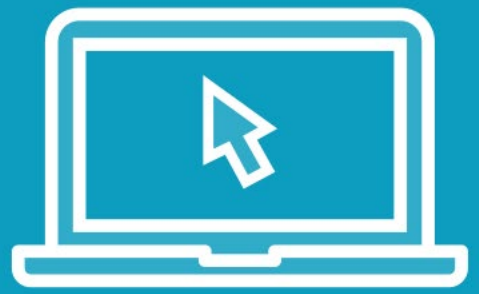
Indexes with far more writes than reads may not be useful for workload

SQL Server must update these indexes as data changes

Unused indexes increase database size and maintenance workload

Make sure you have seen your complete workload before dropping indexes

Demo



Bad NC Indexes

Missing Indexes

Missing indexes for current database

This query is very useful but easy to misinterpret

Do careful analysis before adding new indexes

Missing Indexes

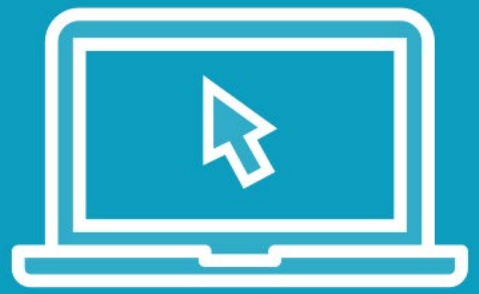
Look at all of the columns returned by this query

Know how long SQL Server has been running as you interpret results

Pay special attention to “last_user_seek”, “user_seeks”, and “avg_total_user_cost columns”

Consider existing indexes and try to create fewer, wider indexes

Demo



Missing Indexes

Missing Index Warnings

Finds missing index warnings in plan cache

Query can take a long time to return results

Returns object name and query plan

Missing Index Warnings

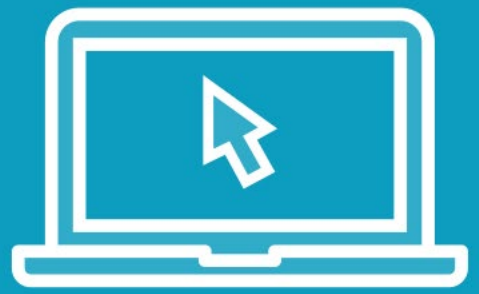
Can associate missing index requests with specific stored procedures

“Usecounts” column shows count of times index was requested by SP/query

Execution plan will have missing index details

Knowing which stored procedure is generating request helps you make better tuning decisions

Demo



Missing Index Warnings

Overall Index Usage – Reads

Shows which indexes in current DB have most reads

Helps you understand your workload

Index reads are beneficial for SELECT query performance

Overall Index Usage – Reads

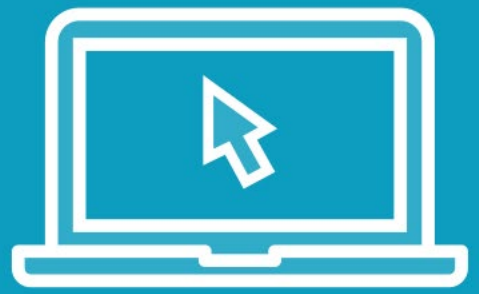
Indexes with high reads may benefit from data compression

Evaluate data volatility and compressibility

Tables with extremely high reads might be columnstore index candidates

Returns cumulative metrics for all row-store indexes in current database

Demo



Overall Index Usage – Reads

Overall Index Usage – Writes

Shows which indexes in current DB have most writes

Helps you understand your workload

Index writes are bad for INSERT / UPDATE query performance

Overall Index Usage – Writes

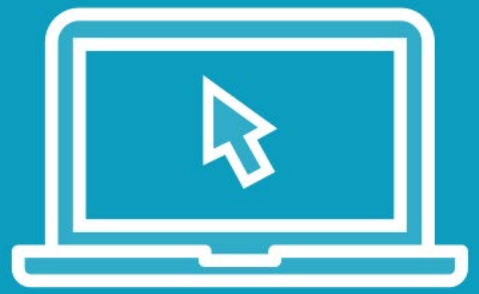
Look for indexes with many more writes than reads

Make sure you know how long SQL Server has been running

Do not blindly drop indexes without more analysis

Returns cumulative metrics for all row-store indexes in current database

Demo



Overall Index Usage – Writes

Volatile Indexes

Shows which indexes and statistics have most updates

Helps you understand your write workload

Helps you design and configure your storage

Volatile Indexes

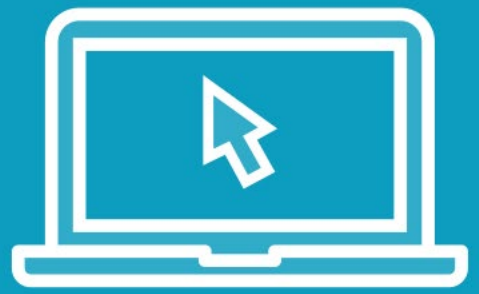
Be more cautious about creating new indexes on volatile tables

Be more cautious about using data compression on volatile tables

Consider moving highly volatile tables/indexes to separate file group

Consider using flash storage or non-parity RAID levels for volatile data

Demo



Volatile Indexes

What We Covered

- Diagnosing activity-related database issues
- Interpreting query results
- Alleviating activity-related database issues