Diagnosing Activity-related Instance Issues

Glenn Berry
PRINCIPAL CONSULTANT – SQLSKILLS.COM

@GlennAlanBerry www.sqlskills.com/blogs/glenn

Module Overview

Activity-related instance issue queries

Interpreting the results of these queries

Version Information

SQL Server version and edition information

Operating system version and edition

Hypervisor presence

Version Information

Shows SQL Server version, edition information, and exact build number

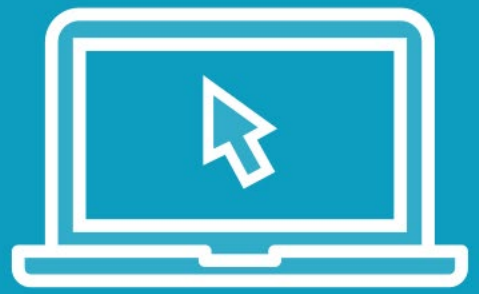
Shows operating system version and edition information

Helps you understand features and capabilities of your instance

Helps know if your SQL Server instance has been properly patched

Knowing if a hypervisor is present is useful but doesn't mean SQL Server is running in the hypervisor

Demo

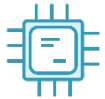


Version Information

Hardware Information

- Processor core counts
- Memory information
- NUMA nodes and soft NUMA
- SQL Server memory model

Hardware Information



Provides great overview of processor and memory configuration

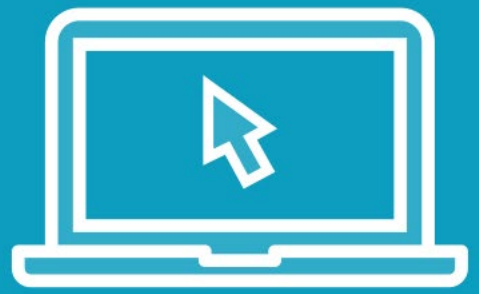
Helps you understand what type of basic environment you are running on

SQL Server 2017 added several new columns of useful information

This information is applicable for both VMs and bare physical machines

Don’t be deceived by the “Virtual Machine Type” column!

Demo



Hardware Information

Average Task Counts

Gives quick overview of instance activity

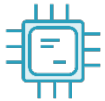
Values will change from second to second

Quickly reveals area of most pain

Average Task Counts

“Avg Task Count” relates to magnitude of workload or locking/blocking

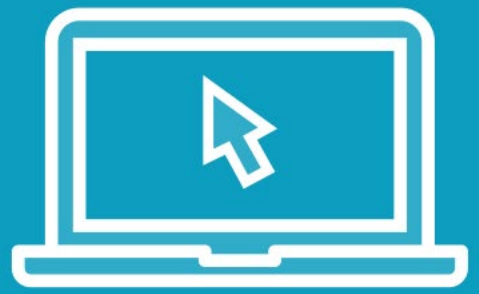
“Avg Work Queue Count” reflects magnitude of workload



“Avg Runnable Task Count” represents tasks waiting for CPU time

“Avg Pending DiskIO Count” represents tasks waiting for I/O to complete

Demo



Average Task Counts

Top Waits

Shows cumulative top wait types

Helps direct your analysis to a more specific area

Wait statistics shown are since last service start or wait statistics clearing

Most useful when instance is under a heavy load

Top Waits

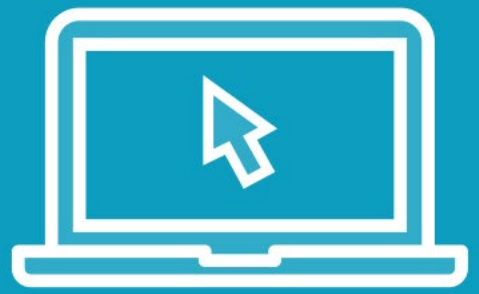
This can be extremely useful information, but use it with care!

Do not do “knee-jerk” performance tuning based solely on wait statistics

Gather more information and do thoughtful analysis before you make changes

Wait statistics are not as useful on a relatively idle instance

Demo



Top Waits

Detect Blocking

Can reveal key information about blocking

Only useful while blocking is actually happening

Run multiple times to get complete information

Detect Blocking

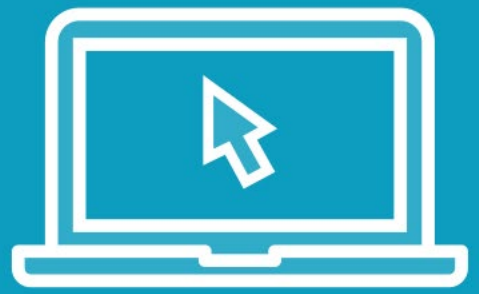
This gives you very useful information when blocking is occurring

Shows lock type, database, SIDs for blocker and waiter, and blocker and waiter batch

The results from this query can be very dynamic on a busy server

This query helps you understand what might be causing blocking

Demo



Detect Blocking

Version Store Space Usage

Returns version store space usage in tempdb

Results are aggregated by database

Returns reserved page count and reserved MB space usage

Version Store Space Usage

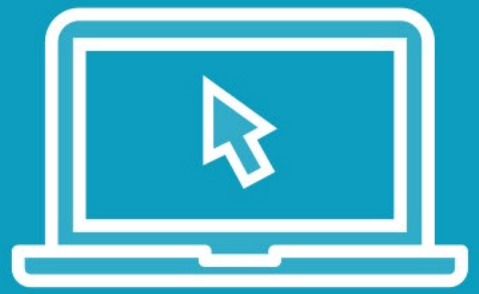
Shows which databases are using version store space in tempdb

Especially useful when your databases use snapshot isolation or RCSI

Helps understand which workloads are using space in tempdb

Also helps you evaluate your tempdb size requirements

Demo



Version Store Space Usage

Top Average Elapsed Time Queries

Returns metrics about highest average elapsed time queries

Can often reveal easy query tuning opportunities

Pay attention to missing index warning column

Concentrate on top five results first

Top Average Elapsed Time Queries

Elapsed times are in microseconds, not milliseconds

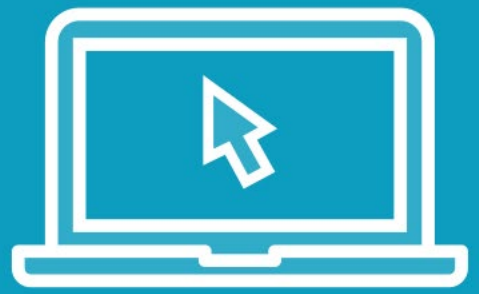
Look for wide variations between min, average, and max elapsed times

Look at execution count to help evaluate importance of the query

Consider other metrics to help evaluate importance of the query

Look at the graphical execution plan for top queries

Demo



Top Average Elapsed Time Queries

UDF Statistics by Database

Returns info on scalar UDF usage

Scalar UDFs are well-known performance issue

Consider refactoring scalar UDFs

UDF Statistics by Database

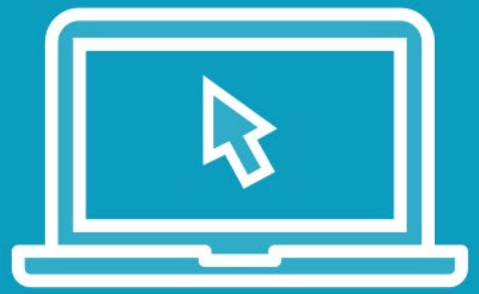
Scalar UDF code gets executed for every row in the resultset

Consider in-lining scalar UDF code if possible

Converting to a table-valued UDF that returns one value

Convert scalar UDF code to a T-SQL stored procedure

Demo



UDF Statistics by Database

Missing Indexes for All Databases

Very useful query for index tuning

Do not over-index based on the results of this query!

You need to understand how long the instance has been running

Understand your workload and the existing indexes

Missing Indexes for All Databases

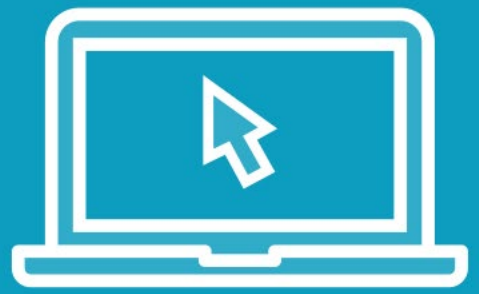
Returns missing index suggestions for all databases on the instance

It is better to create fewer, wider indexes rather than more, narrower indexes

Carefully consider “avg_total_user_cost”, “avg_user_impact” and “user_seeks”

Consider volatility of table as you create additional indexes

Demo



Missing Indexes for All Databases

Connection Counts by IP Address

Good indicator of relative workload

Useful for troubleshooting connectivity issues

Shows where your connections are coming from

Connection Counts by IP Address

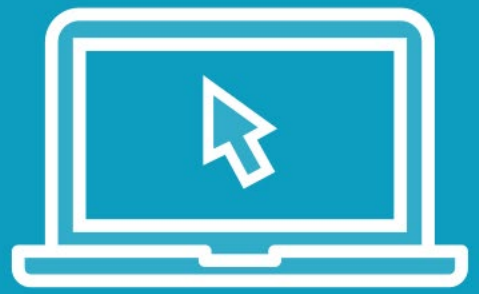
Aggregates connection counts by IP address

Returns “program_name” for connection

Returns “host_name” for connection

Returns “login_name” for connection

Demo



Connection Counts by IP Address

What We Covered

Activity-related instance issue queries

Interpreting the results of these queries