# Instance-Level Performance Queries Part 2

Glenn Berry
Glenn@SQLskills.com
GlennAlanBerry

# Instance-Level Performance Queries

- **A group of queries to collect instance-level performance metrics**
  - These can be run in the context of any database on the instance
  - These are not database specific
- **Many SQL Server instances have instance-level performance issues**
  - These queries help you focus your tuning efforts in the right area
- **My Pluralsight course *Scaling SQL Server 2012 – Part 1* covers best practice instance-level performance considerations**
  - http://bit.ly/1iL0NQR
- **Joe Sack's Pluralsight course *SQL Server: Common Performance Issue Patterns* is also a valuable resource**
  - http://bit.ly/1nTzupp

# Connection Counts

- **This DMV query tells you which logins are connected, how they are connected, and how many sessions they have**
  - sys.dm_exec_sessions
  - MSDN link: http://bit.ly/1rKtAEL
- **This helps you characterize your workload intensity**
  - It also helps you determine whether you are seeing a normal level of activity
- **This also helps you troubleshoot connectivity and permissions issues**
  - You can confirm which logins have active connections

# Connection Counts by IP Address

- **This DMV query tells you which logins are connected by IP address, how they are connected, and how many connections they have**
  - sys.dm_exec_sessions
    - MSDN link: http://bit.ly/1rKtAEL
  - sys.dm_exec_connections
    - MSDN link: http://bit.ly/1iP3BeS
- **This helps you characterize your workload intensity**
  - It also helps you determine whether you are seeing a normal level of activity
  - It also helps you confirm connectivity to middle-tier machines

# Average Task Counts

- **This DMV helps you determine how busy your system is**
  - sys.dm_os_schedulers
    - MSDN link: http://bit.ly/1hiPPwt
- **This will show you average counts across all of your schedulers**
  - Avg Task Count – general indicator of the volume of tasks
    - Sustained values above 10 are often an indicator of blocking issues
  - Avg Runnable Task Count – tasks that are waiting on CPU resources
    - Sustained values above 0 are an indicator of CPU pressure
  - Avg Pending DiskIO Count – pending I/Os that are waiting to be completed
    - Sustained values above 0 are an indicator of I/O pressure
- **You need to run this query multiple times on a regular basis**
  - Running it just once will not give you an accurate picture of what is happening with your instance over time

# CPU Utilization History

- **This DMV is related to the operating system**
  - sys.dm_os_ring_buffers
    - MSDN link: This DMV is not documented in MSDN
- **This query gives you CPU utilization history in one minute increments**
  - It only goes back 256 minutes
  - It shows CPU utilization by SQL Server
  - It also shows total CPU utilization by all other processes on the server
  - This helps you determine your average CPU utilization over time
- **You can easily capture this information and write it to a user database**
  - This will give you long-term baseline and trending information

# Top Worker Time Queries

- **This query uses one DMV and two DMFs**
  - sys.dm_exec_query_stats
    - MSDN link: http://bit.ly/1rcBtS7
  - sys.dm_exec_sql_text
    - MSDN link: http://bit.ly/1oJ7Hog
  - sys.dm_exec_query_plan
    - MSDN link: http://bit.ly/1pEZzp3
- **This query shows you the most expensive queries and stored procedures from a CPU perspective across the entire instance**
  - Worker time equates to CPU cost
  - This only shows cached query plans since the last time SQL Server was started
  - Cached query plans can be cleared in several other ways

# System Memory

- **This DMV returns useful information about your system memory state**
  - sys.dm_os_sys_memory
    - MSDN link: http://bit.ly/1kundCE
- **The query returns these columns**
  - Physical Memory (MB)
  - Available Memory (MB)
  - Total Page File (MB)
  - Available Page File (MB)
  - System Cache (MB)
  - System Memory State
    - You want to see “Available physical memory is high”

# Process Memory

- **This DMV returns information about your SQL Server process memory**
  - sys.dm_os_process_memory
    - MSDN link: http://bit.ly/1fD59ZT

- **The query returns these columns**
  - SQL Server Memory Usage (MB)
    - Use this value instead of what Task Manager reports
  - large_page_allocations_kb
  - locked_page_allocations_kb
  - page_fault_count
  - memory_utilization_percentage
  - available_commit_limit_kb
  - process_physical_memory_low
  - process_virtual_memory_low
    - You want to see a 0 for both of these, meaning they are not low

# Page Life Expectancy by NUMA Node

- **This query returns page life expectancy (PLE) by NUMA node**
  - sys.dm_os_performance_counters
    - MSDN link: http://bit.ly/PV4xSZ
- **You will get one row for each NUMA node in your system**
  - Symmetrical multi processor (SMP) systems will only have one row
  - Virtual machines may only have one NUMA node, depending on how they were configured
- **PLE is one way of measuring memory pressure**
  - You need to monitor your normal PLE range and watch it over time
  - Higher values are better than lower values
- **Don’t believe outdated guidance about 300 being a “good” value**
  - That was valid when servers had much less memory than they do now

# Memory Grants Pending

- **This query returns the current number of memory grants pending**
  - sys.dm_os_performance_counters
    - MSDN link: http://bit.ly/PV4xSZ
- **This is a good measure of memory pressure**
  - Any value above 0 is a very strong indicator of memory pressure
  - It is pretty unusual (and bad) to see values above 0
  - You need to run this query multiple times, since the results will change rapidly

# Memory Clerk Usage

- **This is a DMV query that returns information about total memory usage by each memory clerk**
  - sys.dm_os_memory_clerks
    - MSDN link: http://bit.ly/PV4xSZ
- **This helps you see what part of SQL Server is using the most memory**
  - MEMORYCLERK_SQLBUFFERPOOL should be the highest consumer in SQL Server 2012 and SQL Server 2014
  - CACHESTORE_SQLCP is for cached query plans for ad hoc and prepared queries
    - Watch out for high values with this memory clerk

# Ad Hoc Queries

- **This is a DMV and a DMF that return information on cached query plans**
  - sys.dm_exec_cached_plans
    - MSDN link: http://bit.ly/R9d59I (ending in capital i)
  - sys.dm_exec_sql_text
    - MSDN link: http://bit.ly/1rKMxav
- **This helps you find large, single-use cached ad hoc and prepared query plans that are wasting space in the plan cache**
  - Converting those queries to stored procedures or parameterized queries can help improve this issue
  - Using the "optimize for ad hoc workloads" instance-level setting can help
  - You may have to flush the ad hoc and prepared plan cache periodically
    - DBCC FREESYSTEMCACHE ('SQL PLANS')
- **See Kimberly Tripp's Pluralsight course *SQL Server: Optimizing Ad Hoc Statement Performance* at http://bit.ly/II4EO7 (capital i then capital i)**

# Course Summary

- **These queries can detect most instance-level performance issues**
  - They can also help you find instance-level settings that may be incorrect
- **They give you performance information about your processors, memory, and storage subsystem**
- **They can help you find your instance-level bottlenecks that warrant more detailed investigation and troubleshooting**
- **Make sure to also watch the other DMV courses for more queries**
- **Thanks for watching!**