SQL Server 2014: DMV Diagnostic Queries – Part 2

Instance-Level Performance Queries Part 1

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

I/O Warnings

- **This query uses xp_readerrorlog to look for 15 second I/O warnings**
  - It looks in the five most recent SQL Server error logs
  - It may take some time to complete if your error logs are very large
  - Use global trace flag 3226 to help reduce the size of your error logs
- **This query helps you understand your overall I/O performance**
  - Seeing lots of 15 second I/O warnings is a pretty clear sign of poor I/O performance
  - Pay attention to which drives and which database files are involved
  - Pay attention to the time of day when they occur
    - There may be a scheduled task or job that is causing I/O performance issues

Drive-Level Latency

- **This query uses the sys.dm_io_virtual_file_stats DMV joined with sys.master_files**
  - MSDN link: http://bit.ly/1pSQ4YE
  - It returns drive-level read, write, and total latency in milliseconds, along with the average size of reads, writes, and total transfers in bytes
  - It only returns results for drives where you have SQL Server data or log files
- **This query helps you understand your overall drive performance**
  - It shows you the latency information by drive
  - Above 20-25ms is usually considered high latency
  - These results are cumulative, since SQL Server has been running
  - It also helps you characterize your workload from an I/O perspective

I/O Latencies by Database File

- **This query uses the sys.dm_io_virtual_file_stats DMV joined with sys.master_files**
  - MSDN link: http://bit.ly/1pSQ4YE
  - It returns file-level read, write, and total latency in milliseconds, along with the size of the file and numbers of reads, writes, and total transfers
- **This query shows your overall database file I/O performance**
  - It shows you the latency information by database file
  - Above 20-25ms is usually considered high latency
  - It also helps you characterize your workload from an I/O perspective
- **Drive-level latency and file-level latency numbers give you very useful metrics about your storage subsystem**
  - This is also very helpful for discussions with your server administrator or SAN administrator

Database Properties

- **This query gets database properties from sys.databases and uses sys.dm_os_performance_counters to get transaction log usage information**
  - MSDN link: http://bit.ly/PV4xSZ
- **Returns database property information for every user and system database on the current SQL Server instance**
  - Many different database properties are available
    - Log reuse wait description
    - Database compatibility level
    - Page verify option
    - Auto create statistics, auto update statistics, auto update statistics asynchronously
    - Forced parameterization, snapshot isolation, read-committed snapshot
    - Auto close and auto shrink
    - Change data capture
    - Target recovery time
    - Delayed durability

Missing Indexes for All Databases

- **This query returns information about candidate "missing indexes" for all databases on the current instance**
  - sys.dm_db_missing_index_group_stats
    - MSDN link: http://bit.ly/1rU6o94
  - sys.dm_db_missing_index_groups
    - MSDN link: http://bit.ly/1hYbjTh
  - sys.dm_db_missing_index_details
    - MSDN link: http://bit.ly/1pU8yry
- **This query is both useful and potentially dangerous**
  - It can help find missing indexes with very high impacts
  - It encourages less experienced DBAs to over-index their databases
  - It sometimes recommends duplicate indexes
  - Never just blindly create every index that it recommends!

Getting VLF Counts

- **This query calls DBCC LOGINFO for each database on the current instance which returns the number of virtual log files (VLFs) in each database**
- **High numbers of VLFs can make database recovery take much longer than normal**
  - This affects database restore time and fail-over time with traditional failover clustering
  - Can also affect transaction rollback time and anything that must read the log
- **Kimberly Tripp's blog post "Transaction Log VLFs – too many or too few?" has more details**
  - http://bit.ly/19G2nOd

CPU Usage by Database

- **This query uses a DMV and a DMF to return information about CPU usage for each database on the current instance**
  - sys.dm_exec_query_stats
    - MSDN link: http://bit.ly/1ku0974
  - sys.dm_exec_plan_attributes
    - MSDN link: http://bit.ly/1ilMh6g (lowercase i then uppercase i)
- **This tells you which databases are using the most processor resources on the instance**
  - This is important to know if you are seeing signs of CPU pressure
  - It also helps you characterize your workload from a processor perspective

I/O Usage by Database

- **This query uses a DMF to return information about total I/O usage for each database on the current instance**
  - sys.dm_io_virtual_file_stats
    - MSDN link: http://bit.ly/1pSQM8e
- **This tells you which databases are using the most I/O resources on the instance**
  - This is important to know if you are seeing signs of I/O pressure
  - It also helps you characterize your workload from a I/O perspective

Total Buffer Usage by Database

- **This query uses a DMV to return information about total buffer pool usage for each database on the current instance**
  - sys.dm_os_buffer_descriptors
    - MSDN link: http://bit.ly/1ghI0Xs (capital i then zero)
- **This tells you which databases are using the most buffer pool space on the instance**
  - This is important to know if you are seeing signs of memory pressure
  - It also helps you characterize your workload from a memory perspective
  - This query can take some time to complete on a busy instance

Top Waits

- **This DMV query gives you information about your cumulative wait statistics since the instance was started or statistics were cleared**
  - sys.dm_os_wait_stats
    - MSDN link: http://bit.ly/PVeVtY
- **Lets you determine what SQL Server has spent the most time waiting on**
  - This information can help you find the most important bottlenecks
  - It is very easy to misinterpret the information this query returns and do “knee-jerk” performance tuning
  - There is a lot of misinformation about wait types and what they mean
- **Paul Randal’s Pluralsight course *SQL Server: Performance Troubleshooting Using Wait Statistics* is a great resource**
  - http://bit.ly/1fCOZzx

Signal Waits

- **This DMV query gives you information about your signal waits and other resource waits**
  - sys.dm_os_wait_stats
    - MSDN link: http://bit.ly/PVeVtY
- **Signal waits are CPU-related**
  - High signal waits can be a confirming indicator of CPU pressure
  - Signal waits above 15-20% of total wait time is usually a sign of CPU pressure

Summary

- **These DMV/DMO queries can help you detect most instance-level performance issues**
  - They can help you focus your performance troubleshooting efforts in the right area
- **They can help you characterize your overall workload**
  - Which databases are using the most CPU, I/O, and memory resources
  - What type of overall workload you are seeing on the instance
- **You can get a good idea how your instance is running with these queries**
  - Many SQL Server user databases have multiple properties that are set to an inappropriate value, which can affect instance performance
  - Many SQL Server instances have performance issues that you can detect and diagnose with these queries

What Is Next?

- **Module 3: Instance-Level Performance Queries Part 2**
  - Connection counts
  - Connection counts by IP address
  - Average task counts
  - CPU utilization history
  - Top worker time queries
  - System memory
  - Process memory
  - PLE by NUMA node
  - Memory grants pending
  - Memory clerk usage
  - Ad hoc queries