Database-Level Performance Queries
Part 1

Glenn Berry
Glenn@SQLskills.com
GlennAlanBerry

pluralsight
hardcore dev and IT training

Database-Level Performance Queries

- **A group of queries to collect database-level performance metrics**
  - These must be run in the context of the database you are concerned with
  - These are database specific queries
- **Many SQL Server databases have performance issues**
  - These queries help you further 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

File Sizes and Space

- **This query tells you how large and how full your database files are**
  - It uses the sys.database_files catalog view
    - MSDN link: http://bit.ly/1HvKzp0
  - It also uses the sys.data_spaces catalog view
    - MSDN link: http://bit.ly/1xzmUQj
- **It tells you where the files are located and what file group they are in**
  - Log files are not in a file group
  - You don’t want to use percent growth for auto grow

I/O Stats by File

- **This query uses the sys.dm_io_virtual_file_stats DMV joined with sys.database_files**
  - MSDN link: http://bit.ly/1pSQ4YE
  - It returns file-level statistics for numbers of reads and writes, and io stalls in milliseconds, along with the total size of reads and writes in megabytes
  - It only returns results for the files in the current database
- **This query helps you understand your database file performance**
  - It shows you the various read/write metrics by database file
  - These results are cumulative, since SQL Server has been running
  - This helps you characterize your workload from an I/O perspective

Query Execution Counts

- **This query uses the sys.dm_exec_query_stats DMV cross applied with sys.dm_exec_sql_text**
  - sys.dm_exec_query_stats
    - MSDN link: http://bit.ly/1rcBtS7
  - sys.dm_exec_sql_text
    - MSDN link: http://bit.ly/1oJ7Hog
- **This query shows your top queries by execution counts**
  - Shows ad hoc queries and query text inside of stored procedures
  - Shows selected query execution metrics
  - This helps you better understand your query workload
  - It is filtered by the current database

Stored Procedure Execution Counts

- **This query uses the sys.procedures catalog view joined with sys.dm_exec_procedure_stats DMV**
  - sys.procedures
    - MSDN link: http://bit.ly/1F9O3tg
  - sys.dm_exec_procedure_stats
    - MSDN link: http://bit.ly/1p6Hd5S
- **Tells you which cached stored procedures are called the most often**
  - This helps you understand your stored procedure workload
  - It can help identify possible candidates for middle-tier caching

Stored Procedure Average Elapsed Time

- **This query uses the sys.procedures catalog view joined with sys.dm_exec_procedure_stats DMV**
  - sys.procedures
    - MSDN link: http://bit.ly/1F9O3tg
  - sys.dm_exec_procedure_stats
    - MSDN link: http://bit.ly/1p6Hd5S
- **Tells you which cached stored procedures have the highest average elapsed times (in microseconds)**
  - This helps you understand your stored procedure workload
  - It can help identify possible query tuning candidates
    - This where you can sometimes have a large beneficial impact on the workload

Stored Procedure Average Elapsed Variable Time

- **This query uses the sys.procedures catalog view joined with sys.dm_exec_procedure_stats DMV**
  - sys.procedures
    - MSDN link: http://bit.ly/1F9O3tg
  - sys.dm_exec_procedure_stats
    - MSDN link: http://bit.ly/1p6Hd5S
- **Tells you which cached stored procedures have the highest elapsed times (in microseconds)**
  - This helps you understand your stored procedure workload
  - It shows min, average, max, and last elapsed times (in microseconds)
    - Large variances in elapsed times can be an indicator of plan stability problems
  - It can help identify possible query tuning candidates

Stored Procedure Total Worker Time

- **This query uses the sys.procedures catalog view joined with sys.dm_exec_procedure_stats DMV**
  - sys.procedures
    - MSDN link: http://bit.ly/1F9O3tg
  - sys.dm_exec_procedure_stats
    - MSDN link: http://bit.ly/1p6Hd5S
- **Returns stored procedures that have the highest total worker time**
  - Stored procedure worker time relates to CPU cost
  - Shows which SPs are the most expensive from a CPU perspective
  - This also helps you understand your stored procedure workload
  - It can help identify possible tuning candidates if you are under CPU pressure

Stored Procedure Total Logical Reads

- **This query uses the sys.procedures catalog view joined with sys.dm_exec_procedure_stats DMV**
    - sys.procedures
        - MSDN link: http://bit.ly/1F9O3tg
    - sys.dm_exec_procedure_stats
        - MSDN link: http://bit.ly/1p6Hd5S
- **Returns stored procedures that have the highest total logical reads**
    - Stored procedure logical reads relates to memory and read I/O cost (indirectly)
    - Shows which SPs are the most expensive from a memory perspective
    - This also helps you understand your stored procedure workload
    - It can help identify possible tuning candidates if you are under memory pressure

Stored Procedure Total Physical Reads

- **This query uses the sys.procedures catalog view joined with sys.dm_exec_procedure_stats DMV**
  - sys.procedures
    - MSDN link: http://bit.ly/1F9O3tg
  - sys.dm_exec_procedure_stats
    - MSDN link: http://bit.ly/1p6Hd5S
- **Returns stored procedures that have the highest total physical reads**
  - Stored procedure physical reads relates to read I/O cost
  - Shows which SPs are the most expensive from a read I/O perspective
  - This also helps you understand your stored procedure workload
  - It can help identify possible tuning candidates if you are under memory or read I/O pressure

Stored Procedure Total Logical Writes

- **This query uses the sys.procedures catalog view joined with sys.dm_exec_procedure_stats DMV**
  - sys.procedures
    - MSDN link: http://bit.ly/1F9O3tg
  - sys.dm_exec_procedure_stats
    - MSDN link: http://bit.ly/1p6Hd5S
- **Returns stored procedures that have the highest total logical writes**
  - Stored procedure logical writes relates to write I/O cost
  - Shows which SPs are the most expensive from a write I/O perspective
  - This also helps you understand your stored procedure workload
  - It can help identify possible tuning candidates if you are under write I/O pressure

Top I/O Statements

- **This query uses the sys.dm_exec_query_stats DMV cross applied with sys.dm_exec_sql_text**
  - sys.dm_exec_query_stats
    - MSDN link: http://bit.ly/1rcBtS7
  - sys.dm_exec_sql_text
    - MSDN link: http://bit.ly/1oJ7Hog
- **This query shows your top queries by average I/O cost**
  - Shows query text inside of stored procedures
  - Shows query average I/O and execution counts
  - This helps you better understand your query workload from an I/O perspective
  - It is filtered by the current database

Summary

- **These DMV/DMO queries can help you detect most database-level performance issues**
  - They can help you focus your performance troubleshooting efforts in the right area in a particular database
- **They can help you characterize your overall database workload**
  - Which stored procedures are using the most CPU, I/O, and memory resources
  - What type of overall workload you are seeing on the database
- **You can get a good idea how your database is running with these queries**
  - Many SQL Server databases have performance issues that you can detect and diagnose with these queries

What Is Next?

- **Module 3: Database-Level Performance Queries Part 2**
  - Bad non-clustered indexes
  - Missing indexes
  - Missing index warnings
  - Buffer usage
  - Table sizes
  - Table properties
  - Statistics update
  - Index fragmentation
  - Overall index usage - reads
  - Overall index usage - writes
  - Recent full backups