SQL Server: Benchmarking and Baselining

Module 5: Using DMVs

Erin Stellato

Erin@SQLskills.com

pluralsight
hardcore developer training

Introduction

- **Dynamic Management Views (DMVs) and Dynamic Management Functions (DMFs) are built into SQL Server**
    - Collectively known as Dynamic Management Objects (DMOs)
- **Provide information about the server and its databases that can be used to monitor health and performance as well as diagnose problems**
    - Server-scoped
    - Database-scoped
- **Information does not persist between restarts**
    - One exception: sys.dm_db_index_physical_stats
    - In some cases, you can clear data without a restart
- **DMVs have been available since SQL Server 2005**
    - Grouped by functionality
    - Schema changes can occur between versions

When to Use DMVs

- **Easy method to capture, review, and store metrics**
  - No additional utilities
  - Utilize T-SQL
- **Certain information is only available through DMVs**
  - e.g. cached query plans, SQLOS information, wait statistics
- **Very often, dynamic management objects are the best place to start when troubleshooting**
- **Be aware of the overhead generated when querying specific DMVs**
- **Glenn Berry's scripts provide an excellent starting point (http://bit.ly/OB9X4D)**

Capturing DMV Data

- **There are over 175 dynamic management objects in SQL Server 2012**
  - Determine what is most relevant to your environment
- **Snapshot data to a table at regular intervals**
- **Report on captured data as needed**

DMVs to Consider for Data Capture

- **sys.dm_os_sys_info**
  - CPU, memory and SQL Server start time
- **sys.dm_os_sys_memory**
  - Available physical memory, page file, and memory state
- **sys.dm_os_process_memory**
  - Memory currently in use, large page allocations, and whether OS has notified SQL Server that memory is low
- **sys.dm_os_performance_counters**
  - Current value for a performance counter (as discussed in Module 3)
- **sys.dm_os_wait_stats**
  - Aggregated waits for the instance
- **For more information on wait statistics, see the Pluralsight course, SQL Server: Performance Troubleshooting Using Wait Statistics**

DMVs to Consider for Data Capture (2)

- **sys.dm_db_file_space_usage**
  - Lists file size and used space for every database file
  - Works for tempdb only prior to SQL Server 2012
- **sys.dm_io_virtual_file_stats**
  - Reads, writes, latency, and current size for every database file
- **sys.dm_db_index_physical_stats**
  - Size, level of fragmentation, forwarded rows for any index or table
- **sys.dm_db_index_usage_stats**
  - Cumulative seeks, scans, lookups and updates for an index
- **sys.dm_db_missing_index_details**
  - Lists indexes the Query Optimizer has determined are missing
  - Join with sys.dm_db_missing_index_group_stats to understand cost impact

DMVs to Consider for Data Capture (3)

- **sys.dm_exec_requests**
  - Lists queries that are currently executing
- **sys.dm_exec_query_stats**
  - Aggregate statistics for cached query plans including execution count, reads, writes, duration, and number of rows returned
- **sys.dm_exec_procedure_stats**
  - Aggregate statistics for cached stored procedures including execution count, reads, writes, and duration
- **sys.dm_exec_sql_text**
  - Provides the text for a currently executing, or previously executed query, based on plan_handle or sql_handle (commonly obtained from sys.dm_exec_requests)
- **sys.dm_exec_query_plan**
  - Provides the showplan XML for a currently executing, or previously executed query, based on plan_handle (commonly obtained from sys.dm_exec_requests)

Other Data You Can Capture

- **System configuration**
  - sys.configurations, SERVERPROPERTY, DBCC TRACESTATUS, sys.databases
- **Database and file sizes**
  - sys.master_files, sys.database_files, DBCC SQLPERF
- **Database maintenance history**
  - msdb.dbo.backupset, msdb.dbo.sysjobhistory

WhoIsActive

- **Free tool developed by Adam Machanic, SQL Server MVP, which queries multiple DMVs to provide a view of current system performance (http://bit.ly/V6w0Sg)**
- **Extremely useful when troubleshooting performance issues**
- **Can also be utilized to capture baseline information**
- **Data can also be stored to a table on a scheduled basis**
  - Can include query text and query plan for later review

Summary

- DMVs contain integral, valuable information about a SQL Server instance
- As with Performance Monitor counters, the available data can be overwhelming
- DMVs are an excellent way to take baseline data to the next level
- DMVs should also be used to troubleshoot performance issues
- In the next module: Pulling It All Together