

SQLskills Immersion Event IE2: Performance Tuning

Module 10: Memory Performance Tuning

Jonathan Kehayias
Jonathan@SQLskills.com



Overview

- How to identify memory issues
- Internal memory pressure
- External memory pressure
- Virtual address space issues
- Resource semaphores
- DBCC commands



2

© SQLskills, All rights reserved.
<http://www.SQLskills.com>



How to Identify Memory Issues

- **Performance counters**

- Memory\Available Mbytes
- SQLServer:BufferManager\Page Life Expectancy
- SQLServer:BufferManager\Lazy Writes/sec
- SQLServer:BufferManager\Page Reads/sec
- SQLServer:MemoryManager\Memory Grants Pending
- SQLServer:MemoryManager\Memory Grants Outstanding
- Physical Disk\Disk sec/Reads

- **Memory clerks**

- sys.dm_os_memory_clerks
- sys.dm_exec_cached_plans
- sys.dm_os_ring_buffers
- sys.dm_os_memory_cache_clock_hands

How to Identify Memory Issues

- **Wait types**

- CMEMTHREAD
 - Waiting on a thread-safe memory object, generally for inserting or removing execution plans from cache
- RESOURCE_SEMAPHORE
 - Waiting for a workspace memory grant for execution of a sort or hash operation
 - Error 8645 - A time out occurred while waiting for memory resources to execute the query. Rerun the query.
- RESOURCE_SEMAPHORE_QUERY_COMPILE
 - Waiting for a memory grant during query compilation typically due to excessive plan compilations at a given point in time
- RESOURCE_SEMAPHORE_SMALL_QUERY
 - Waiting for a memory grant from the small gateway during execution

Internal Memory Pressure

Buffer Pool

■ Symptoms

- SQLServer:Buffer Manager\Page Life Expectancy low
 - Or on NUMA, SQLServer: Buffer Node X\Page Life Expectancy low
- SQLServer:Buffer Manager\Lazy Writes/sec > 0 outside of CHECKPOINT operations
- SQLServer:Buffer Manager\Free List Stalls > 0
- SQLServer:Buffer Manager\Page Reads/sec excessive
- SQLServer:Memory Manager\Target Server Memory (KB) > Total Server Memory(KB)
- Physical Disk\Disk sec/Reads > 30

Internal Memory Pressure

Buffer Pool

■ Causes

- High data cache churn inside of the SQL Server instance due to workload
- Excessive table or index scans due to inefficient or missing indexes
- Excess index fragmentation or indexes with low page density due to incorrect fill factor specifications
- Implicit conversion/probe residual issues in the workload causing excess Scan operations
- Plan cache bloat due to ad hoc non-parameterized workloads

■ Resolution

- Tune index usage in the environment
- Perform analysis based index maintenance and adjust fill factor based on analysis of index fragmentation ratios
- Remove implicit conversions/probe residual from column side of filtering predicates
- Add memory

Internal Memory Pressure Plan Cache

■ Symptoms

- ❑ SQLServer:Plan Cache\Cache Hit Ratio < 80
- ❑ SQLServer:SQL Statistics\Compilations/sec to SQLServer:SQL Statistics\Batch Requests/sec exceeds 1 compile per 100 batch requests
- ❑ High numbers of single use plans in cache

■ Causes

- ❑ Ad hoc, non-parameterized workload generated by the application
- ❑ Not enough memory to support the workload being generated
- ❑ Auto-Close database option set to true
- ❑ 32-bit SQL Server memory limitations (the plan cache can not use AWE mapped memory)

Internal Memory Pressure Plan Cache

■ Resolution

- ❑ Set Auto-Close database option to false to prevent cache store flushing from occurring
- ❑ Use 'optimize for ad hoc workloads' sp_configure option
- ❑ Convert application to use parameterized calls with fixed SqlDbType parameters
- ❑ Convert application to use stored procedures
- ❑ Add more memory to the server, or increase 'max server memory' in 64-bit environments

■ Reference

- ❑ <http://www.sqlskills.com/BLOGS/KIMBERLY/post/Procedure-cache-and-optimizing-for-adhoc-workloads.aspx> (<http://bit.ly/9MhQTt>)
- ❑ <http://www.sqlskills.com/BLOGS/KIMBERLY/post/Plan-cache-adhoc-workloads-and-clearing-the-single-use-plan-cache-bloat.aspx> (<http://bit.ly/ae4RRI>)

Internal Memory Pressure

Execution Memory

- **Symptoms**

- SQLServer:Memory Manager\Memory Grants Outstanding low
- SQLServer:Memory Manager\Memory Grants Pending high
- RESOURCE_SEMAPHORE wait types on executing tasks

- **Causes**

- Missing indexes resulting in hash and sort operations that consume large amounts of memory
- Insufficient execution memory for current workload in the environment
- Incorrect configuration of Resource Governor

- **Resolution**

- If using Resource Governor, configure the resource pool memory limits higher for the workload
- Tune queries to reduce the execution memory grant requirements

Internal Memory Pressure

Execution Memory (Semaphores)

- **Track resource usage and availability**

- **Prevent race and deadlock conditions**

- **Library analogy**

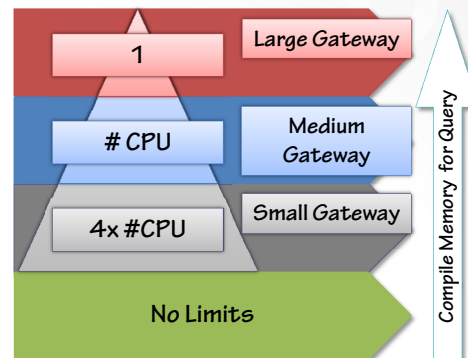
- 10 study rooms available
- Librarian tracks the number of available rooms
- Students request rooms for use and return them when finished
- If no room is available students wait in a queue for the next available room

Internal Memory Pressure Execution Memory (Semaphores)

- Allows a query to reserve memory only if there is enough free memory available
- For a new request, it first checks if any query is waiting
 - If a query is already waiting, the new query is placed into a queue
 - The wait queue is implemented as FIFO queue
 - Small queries queue into the Small Resource queue favoring > 5MB memory requirements
 - If it does not find any waiting query, it then checks available free memory
 - If it does not find enough free memory, then it puts the current query into the waiting queue
 - If it finds enough free memory, then the requested memory is granted and the query can start running
- Grant attempts are made when there is no waiting query, or when an existing query returns reserved memory
- Resource Semaphore wakes up queries in the waiting queue when enough free memory becomes available

Internal Memory Pressure Compile Memory (Gateways)

- Queries begin compiling with no memory limitations
- Once a query passes the compile memory threshold for each gateway, it must obtain a semaphore from the gateway to continue executing
 - If the gateway is full the query will wait with RESOURCE_SEMAPHORE_QUERY_COMPILE until the memory is available at the gateway
 - If the query requires a larger gateway, it maintains its semaphore grant from the lower gateway(s) as well
- Small gateway has fixed threshold
 - x86 – 250,000 bytes
 - x64 – 380,000 bytes
- Medium and large gateways are dynamic based on load



Internal Memory Pressure

Compile Memory

- **Symptoms**
 - RESOURCE_SEMAPHORE_QUERY_COMPILE wait type on executing tasks
- **Causes**
 - Incorrect configuration of Resource Governor
 - Ad hoc workloads causing high compilations
- **Resolution**
 - Review Resource Governor configuration and impact to memory gateway thresholds
 - Each Resource Governor resource pool allows separate large compilations to occur but reduces the gateway threshold
 - Could cause medium gateway queries to become large and compile serially
 - Rewrite queries to reduce compilation memory requirements
 - Improve plan cache reuse through parameterization, stored procedure usage
 - Reduce compilations by using plan guides for common ad hoc queries

External Memory Pressure

Windows OS Pressure

- **Symptoms**
 - SQL Server memory usage constantly changing
 - Memory\Available Mbytes < 150-300MB
 - RING_BUFFER_RESOURCE_MONITOR entries in sys.dm_os_ring_buffers of type RESOURCE_MEMPHYSICAL_LOW
- **Causes**
 - Concurrent application loads on the server with incorrectly configured 'max server memory'
 - Large system cache issues or VM ballooning (future slides)
- **Resolution**
 - Configure 'max server memory' to reserve memory for the OS and other applications

External Memory Pressure

Windows System Cache

- Windows caches buffered I/O operations in the System Cache (e.g. file-copy operations)
- Working set trim problems on 64-bit systems
 - 2GB VAS limit constrains system cache usage on 32-bit
- Dynamic cache service can be used to limit the size of the System Cache
- <http://support.microsoft.com/kb/976618>

External Memory Pressure

Working Set Trims

- Symptoms
 - Slow response times, high CPU, and high disk activity on the page file drive
 - SQL Server 2005 SP2+ error log entry "A significant part of SQL Server process memory has been paged out"
- Causes
 - Incorrect settings for the max server memory sp_configure option, when Lock Pages in Memory is not being used
 - Large system cache in Windows caused by caching of non-buffered I/O operations such as file copy operations
 - Hardware-driver issues that result in memory leaks or excessive memory allocations by the driver

External Memory Pressure

Working Set Trims

Resolution

- ❑ Configure 'max server memory' appropriately to leave memory available for the OS as well as any other applications running on the server
- ❑ Track down the faulty drivers
- ❑ Install the Dynamic Caching Service to minimize the system cache size
- ❑ Remove file shares from the SQL Server and create a separate file server if appropriate
- ❑ Change file copy operations to use a tool that doesn't buffer the I/O
 - ❑ `eseutil /y <srcfile> /d <destfile>` (<http://tinyurl.com/eseutil>)
 - ❑ `XCOPY /J` (Windows Server 2008 R2)
- ❑ Use Lock Pages in Memory in for SQL Server instance

Reference:

- ❑ <http://support.microsoft.com/kb/918483>



17

© SQLskills. All rights reserved.
<http://www.SQLskills.com>

External Memory Pressure

VMware Ballooning

Symptoms

- ❑ Slow response times with high disk I/O on the data drives
- ❑ Windows OS Available Memory < 150MB
- ❑ SQL Server memory usage less than server configuration
- ❑ VMware Memory\Memory Ballooned > 0

Causes

- ❑ Incorrect configuration of the SQL VM in ESX hypervisor for memory overcommit usage in VMware
- ❑ Hypervisor memory pressure resulting in VM memory ballooning to maintain hypervisor stability

Resolution

- ❑ Set min server memory values for SQL Server process inside the VM
- ❑ Set static memory reservations for the SQL Server VM in the hypervisor

Reference: <http://bit.ly/VMwareSQLBalloon>



18

© SQLskills. All rights reserved.
<http://www.SQLskills.com>

Virtual Address Space Issues

▪ Symptoms

- Error 701 FAILED_VIRTUAL_RESERVE
- CLR App Domain Unload due to memory pressure
- Failed backups
- 32-bit environment

▪ Causes

- Limited Virtual Address Space inside of a 32-bit process
- Fragmentation of memory regions due to allocation and de-allocation over time (all VAS allocations must be contiguous)
- Excessive memory requirements (ex. backups with 4MB MAXTRANSFERSIZE and 16 BUFFERCOUNT would require a 64MB contiguous region of VAS)

▪ Resolution

- Migrate to 64-bit installation
- Remove/reduce VAS consumption
 - Lower BUFFERCOUNT and MAXTRANSFERSIZE for backups
 - Tune CLR code to efficiently use memory
- Increase startup reservation with ;g<size> startup parameter



19

© SQLskills. All rights reserved.
<http://www.SQLskills.com>

DBCC Commands

▪ FREEPROCCACHE ({ plan_handle | sql_handle | pool_name })

- Clears the procedure cache of all plans or a single plan if specified
- Often misused as a brute force fix for parameter sniffing issues (this is not the correct way to deal with bad plans)

▪ FREESYSTEMCACHE ({'ALL' | cache_name } [, pool_name])

- Clears all entries for the specified cache from memory, or optionally for a single pool
- Use WITH MARK_IN_USE_FOR_REMOVAL to asynchronously free the currently used cache entries as soon as they are free

▪ FREESESSIONCACHE

- Flushes the distributed query connection cache used by distributed queries against an instance of Microsoft SQL Server.

▪ DROPCLEANBUFFERS

- Clears the buffer pool of non-dirty data pages forcing physical reads from disk (useful for testing performance but not for production)



20

© SQLskills. All rights reserved.
<http://www.SQLskills.com>

Review

- How to identify memory issues
- Internal memory pressure
- External memory pressure
- Virtual address space issues
- Resource semaphores
- DBCC commands

Questions?