Microsoft

SQL Server Processor and Memory Management

Module 2

Learning Units covered in this Module

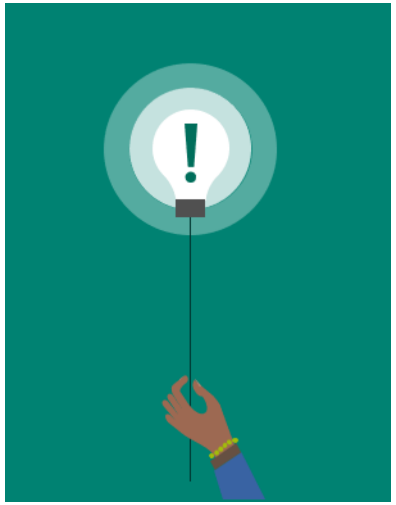
- Lesson 1: Processor Management
- Lesson 2: Non-Uniform Memory Access (NUMA)
- Lesson 3: SQL Server Memory Management
- Lesson 4: Buffer Pool Management
- Lesson 5: Troubleshooting SQL Server memory

Lesson 1: Processor Management

Objectives

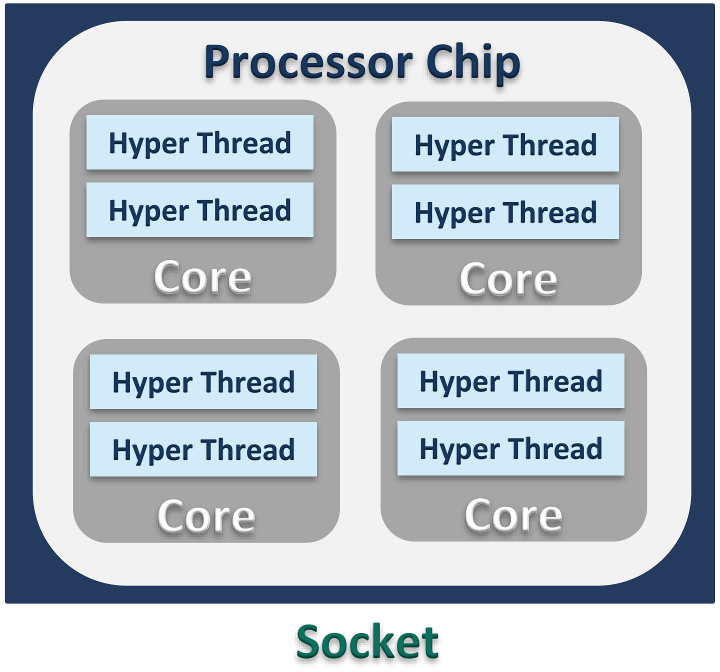
After completing this learning, you will be able to:

- Understand CPU Architecture
- Processor Configuration Settings



CPU Architecture

Physical Hardware



Processor Chip

Hyper Thread
Hyper Thread
Core

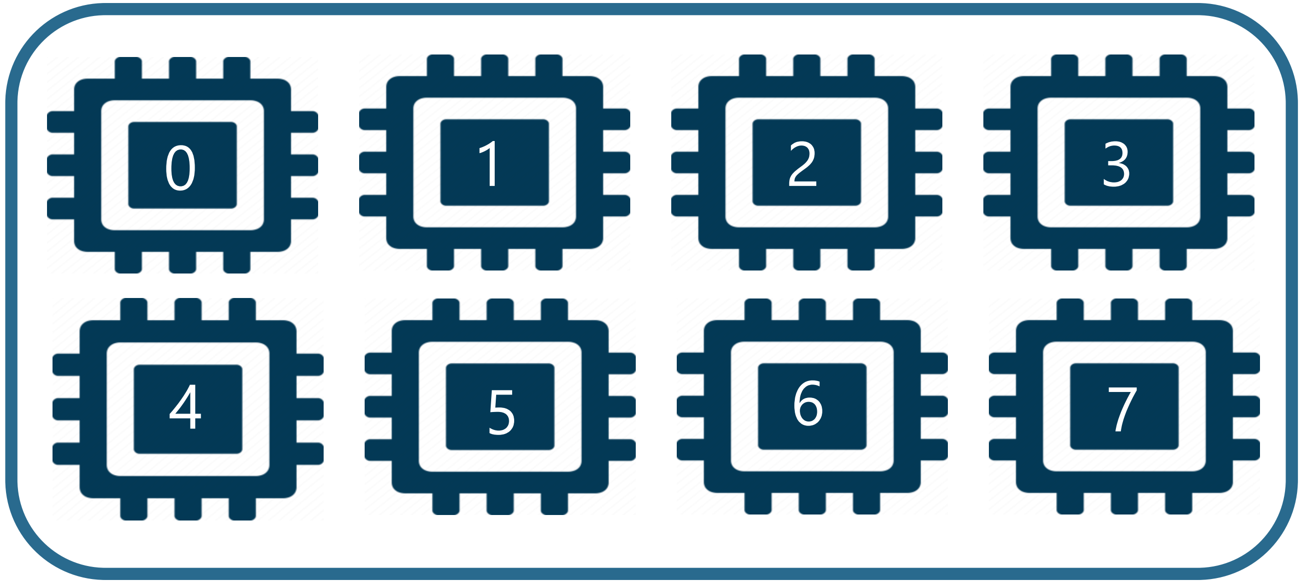
Hyper Thread
Hyper Thread
Core

Hyper Thread
Hyper Thread
Core

Hyper Thread
Hyper Thread
Core

Socket

Logical Processors as seen by the OS



0 1 2 3
4 5 6 7

SQL Server Configuration

Processor Configuration Settings And Best Practices

Affinity Mask

- Assigns CPUs for SQL Server use
- Set via sp_configure or Alter Server Configuration
- Only required in specific scenarios

Max Degree of Parallelism (MAXDOP)

- Maximum number of processors that are used for the execution of a query in a parallel plan. This option determines the number of threads that are used for the query plan operators that perform the work in parallel.

Cost Threshold for Parallelism

- Only queries with a cost that is higher than this value will be considered for parallelism
- Only required when dealing with excessive parallelism

Max Worker Threads

- Number of threads SQL Server can allocate
- Recommended value is 0. SQL Server will dynamically set the Max based on CPUs and CPU architecture. (512 + (Processors -4) *16)

SQL Server Configuration

MAXDOP Setting and Best Practices

Best Practice Recommendations (documented in): KB 2806535

Server with single NUMA node	Less than or equal to 8 logical processors	Keep MAXDOP at or below # of logical processors
Server with single NUMA node	Greater than 8 logical processors	Keep MAXDOP at 8
Server with multiple NUMA nodes	Less than or equal to 16 logical processors per NUMA node	Keep MAXDOP at or below # of logical processors per NUMA node
Server with multiple NUMA nodes	Greater than 16 logical processors per NUMA node	Keep MAXDOP at half the number of logical processors per NUMA node with a MAX value of 16

Notable Waits

CPU related waits

SOS_SCHEDULER_YIELD

- Normally means a thread has yielded after exhausting the 4ms quantum.

THREADPOOL

- Look for high blocking or contention problems with workers.
- This will not show up in sys.dm_exec_requests.
- Might indicate CPU pressure if very high overall percentage of Processor Time. Example: Large amount of Signal Waits (Runnable Queue)

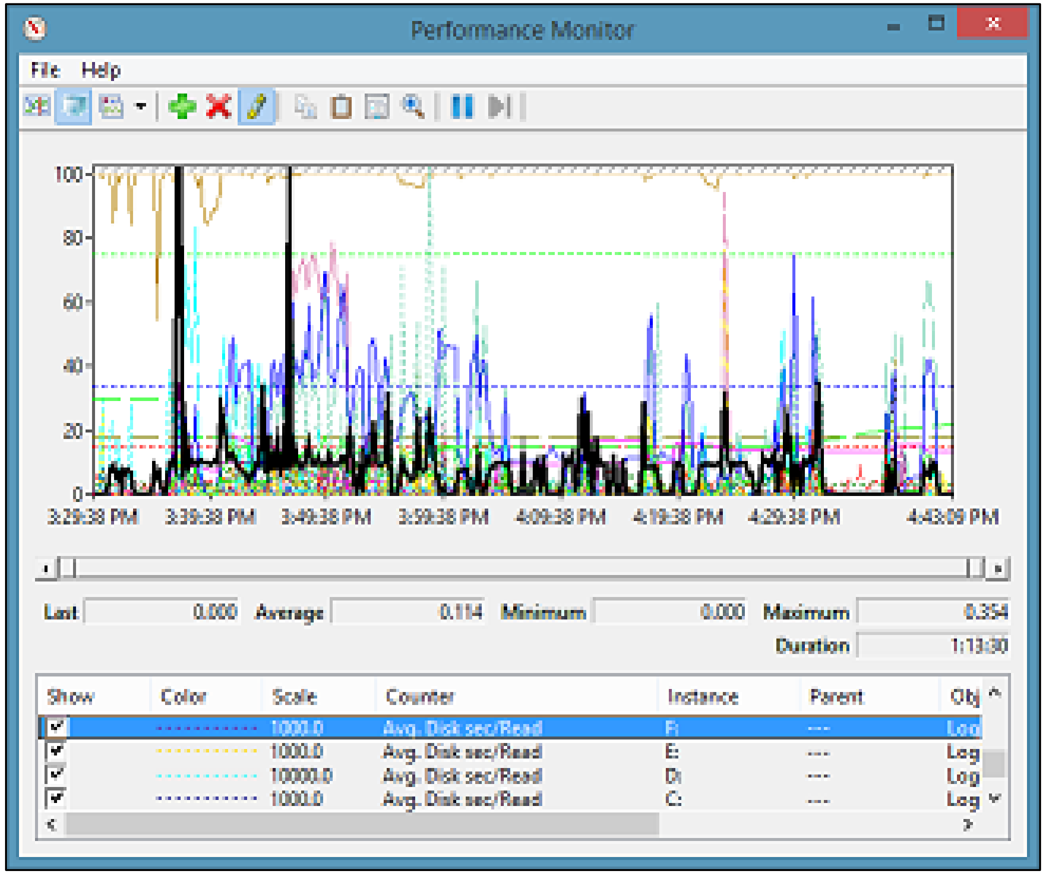
CXPACKET

- If it's an OLTP system, check for parallelism issues if above 20%
- If combined with a high number of PAGEIOLATCH_xx waits, it could be due to large parallel table scans going on because of incorrect non-clustered indexes, or out-of-date statistics causing a bad query plan

Performance Monitor Counters

Important Operating System Counters

- % Processor Time
  - Less than 80% is preferred



Last 0.000 Average 0.114 Minimum 0.000 Maximum 0.354
Duration 1:13:30

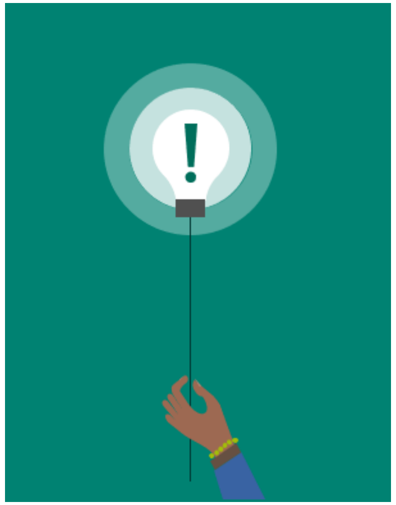
Show	Color	Scale	Counter	Instance	Parent	Obj
☑	············	1000.0	Avg. Disk sec/Read	F:	---	Log
☑	············	1000.0	Avg. Disk sec/Read	E:	---	Log
☑	············	10000.0	Avg. Disk sec/Read	D:	---	Log
☑	············	1000.0	Avg. Disk sec/Read	C:	---	Log

Lesson 2: Non-Uniform Memory Access (NUMA)

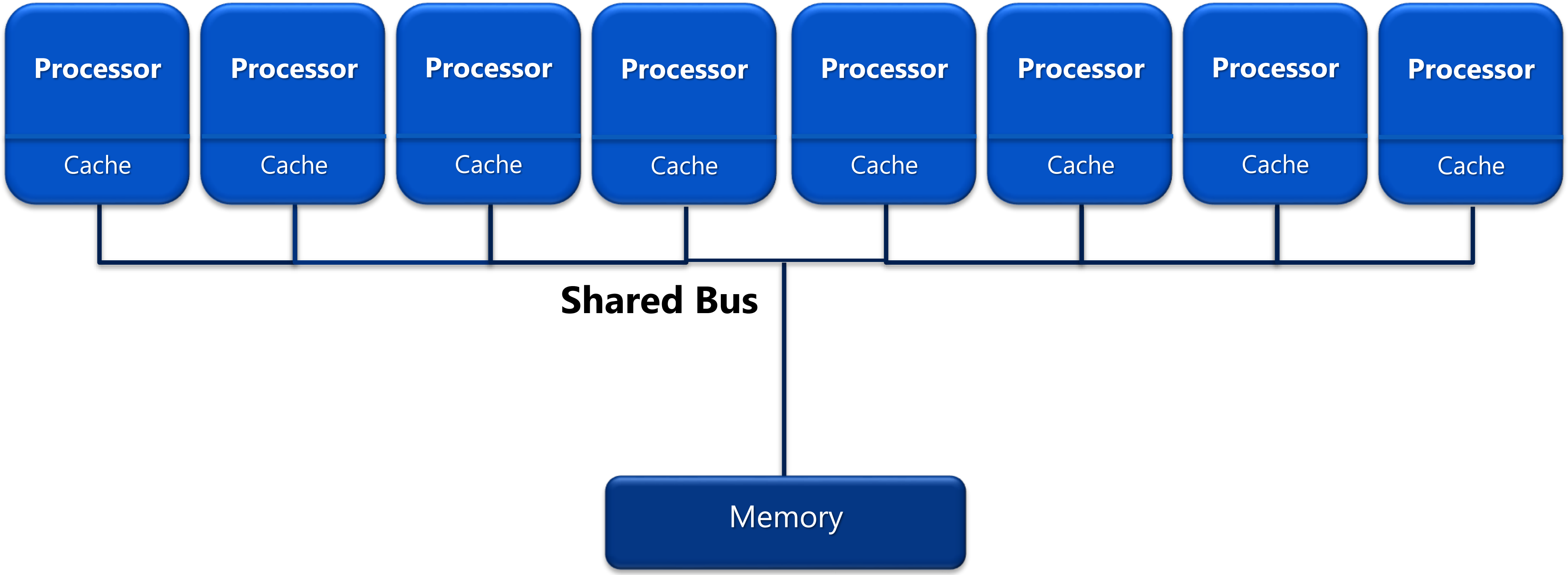
Objectives

After completing this learning, you will be able to:

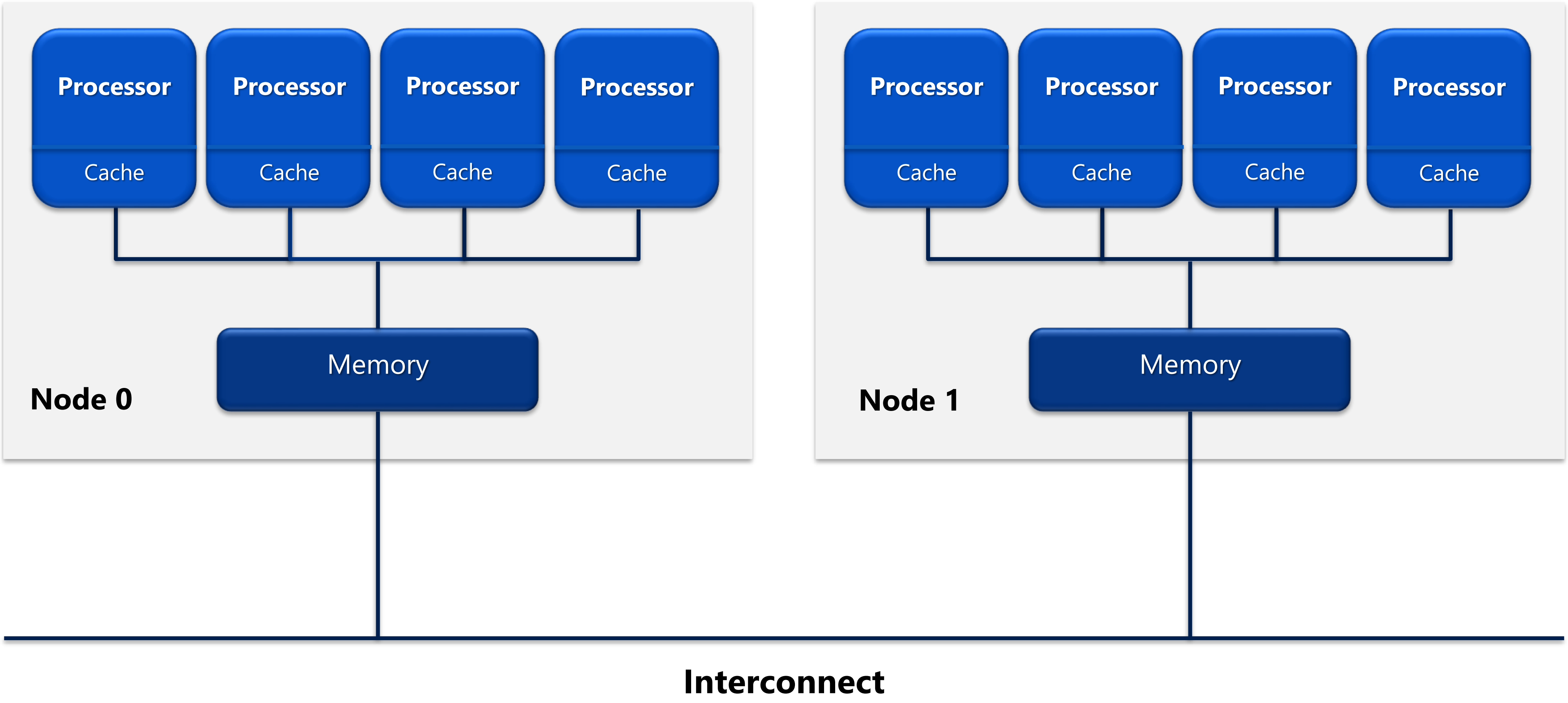
- Understanding Symmetric Multi-Processing (SMP)
- Configuring Non-Uniform Memory Access (NUMA)



Symmetric Multi-Processing (SMP)



Non-Uniform Memory Access (NUMA)



NUMA (Non-Uniform Memory Access) Architecture

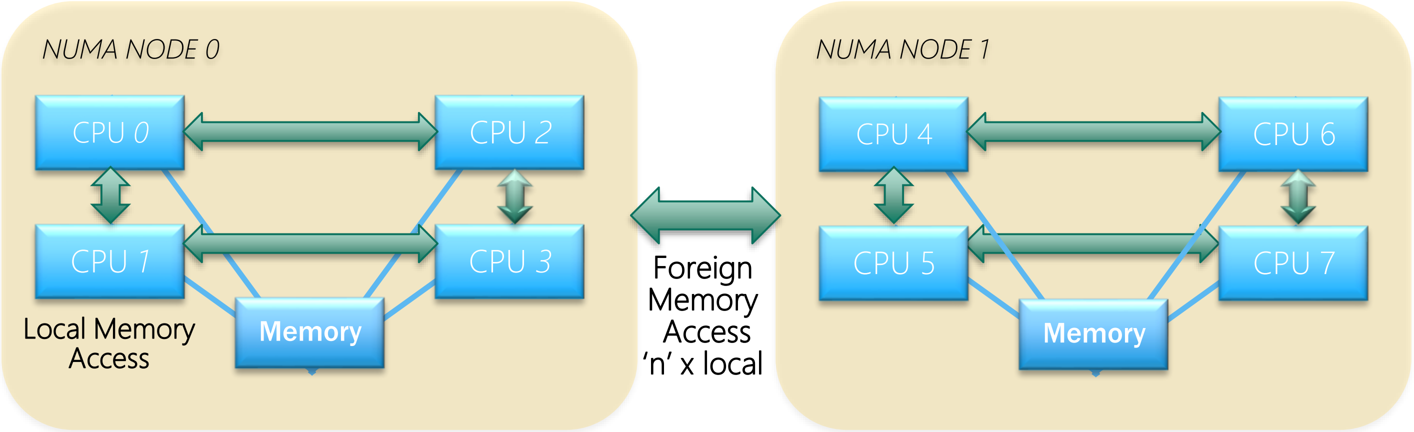
Offers nodes of processors each with its own bus for access for local memory.

Interconnect between nodes allows one node to get to other’s memory.

Offers scalability for NUMA-aware applications.

NUMA-aware applications such as SQL Server try to avoid remote or foreign memory access.

Non-Uniform Memory Access (NUMA)



‘n’ >3 (typically) but varies with hardware design

Automatic Soft NUMA

For systems reporting eight or more CPUs per NUMA node.

At startup, SQL Server 2016 interrogates the hardware layout and automatically configures Soft NUMA.

The Automatic Soft NUMA logic considers logical CPU ratios, total CPU counts and other factors, attempting to create soft, logical nodes containing 8 or fewer CPUs each.

It can provide a gain of up to 20%.

Automatic Soft NUMA (SQL Server 2016)

Automatic Soft NUMA is Hyperthreaded-aware

Check the errorlog for:

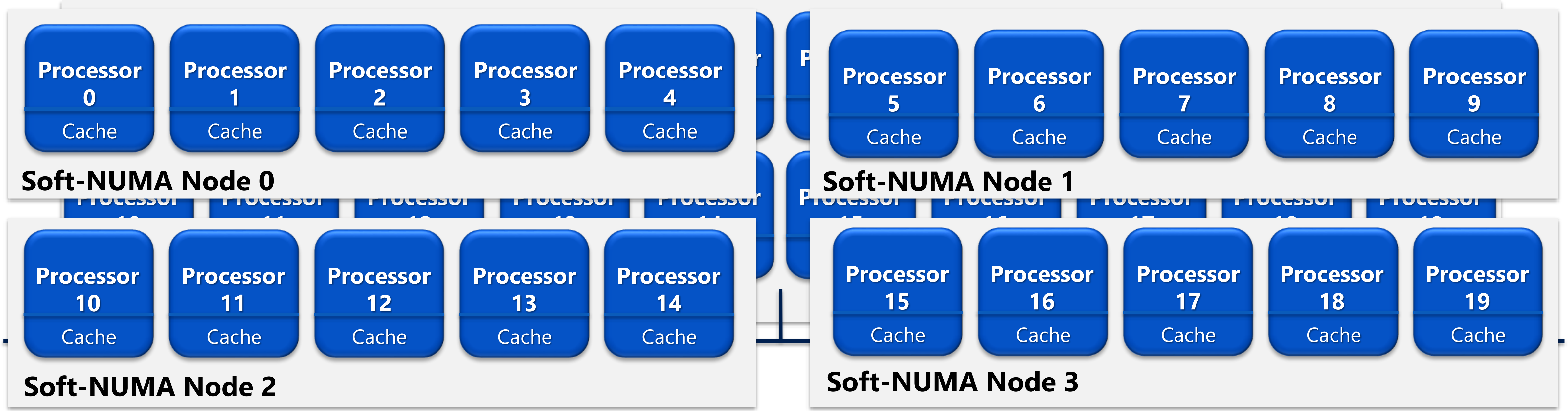
Automatic soft-NUMA was enabled because SQL Server has detected hardware NUMA nodes with greater than 8 logical processors.

Check **softnuma_configuration_desc** column in sys.dm_os_sys_info for one of the three values: **OFF / ON / MANUAL**

Automatic Soft NUMA (SQL Server 2016)

With Soft-NUMA

With Default NUMA



Processor 0 Cache

Processor 1 Cache

Processor 2 Cache

Processor 3 Cache

Processor 4 Cache

Soft-NUMA Node 0

Processor 5 Cache

Processor 6 Cache

Processor 7 Cache

Processor 8 Cache

Processor 9 Cache

Soft-NUMA Node 1

Processor 10 Cache

Processor 11 Cache

Processor 12 Cache

Processor 13 Cache

Processor 14 Cache

Soft-NUMA Node 2

Processor 15 Cache

Processor 16 Cache

Processor 17 Cache

Processor 18 Cache

Processor 19 Cache

Soft-NUMA Node 3

Demonstration

Examining NUMA

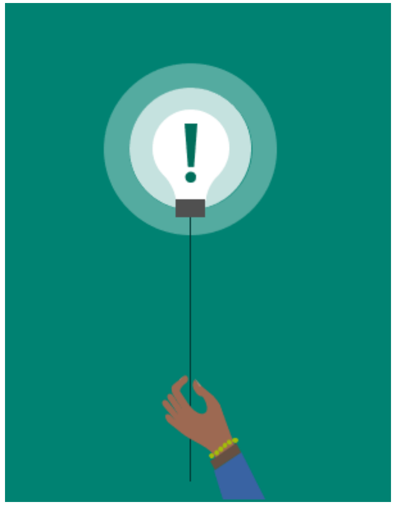
Questions?

Lesson 3: SQL Server Memory Management

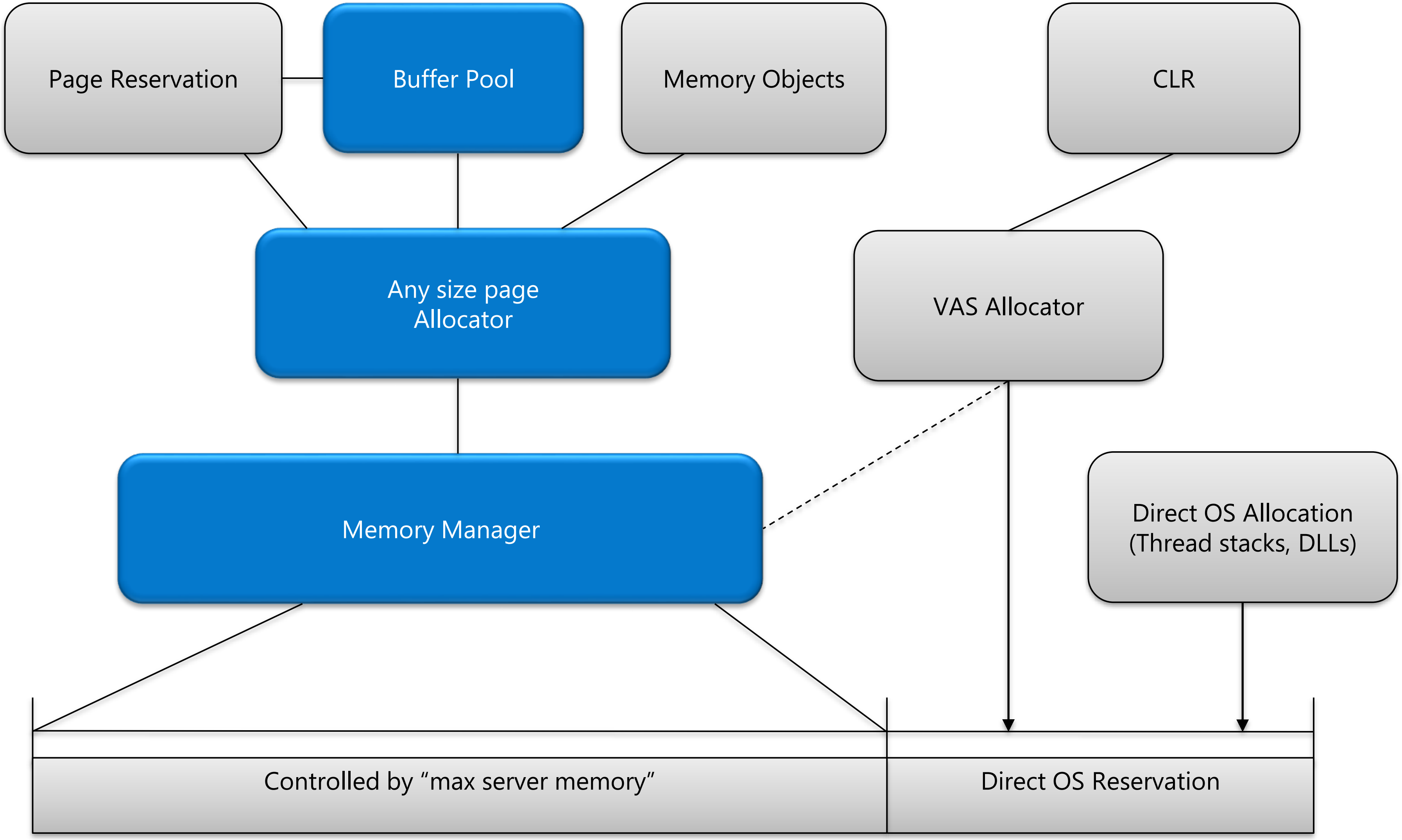
Objectives

After completing this learning, you will be able to:

Understand SQL Server memory management



Memory Manager SQL Server 2012 and later



SQL Server Memory Components

Memory Brokers – Distributes memory across different components based on demands

Memory Clerks – Component that accesses memory node interfaces for allocating memory

Memory Nodes (NUMA) – Memory that represents how memory is consumed across NUMA nodes

Memory Pools (Resource Governor) – Memory that represents how memory is organized by Resource Governor pool / workload group (internal for background and default for user)

Dynamic Memory Management

Max Server Memory defines the maximum amount of memory the SQL Server process can allocate.

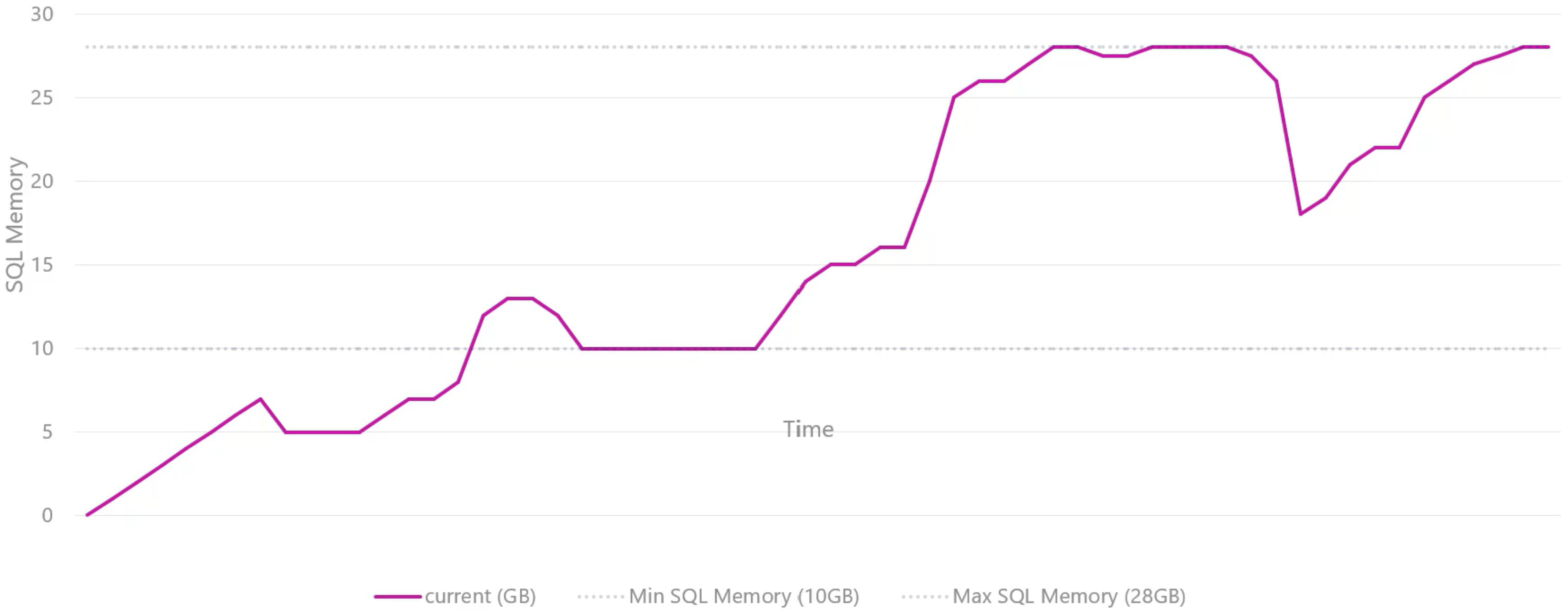
Min Server Memory defines the level down to which SQL Server may trim in the event of memory pressure.

Memory is allocated as-needed after the SQL Server service startup.

After memory is acquired, it will not be released unless the operating system reports memory pressure.

In the event of pressure, SQL Server will reduce its memory footprint to avoid operating system paging.

Dynamic Memory Management



Demonstration

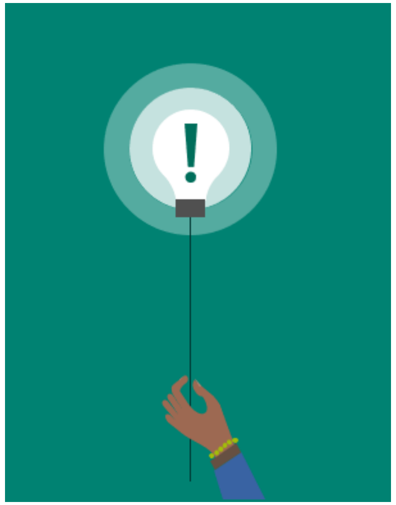
Monitoring Memory Usage

Lesson 4: Buffer Pool Management

Objectives

After completing this learning, you will be able to:

Understand SQL Server Buffer Pool management



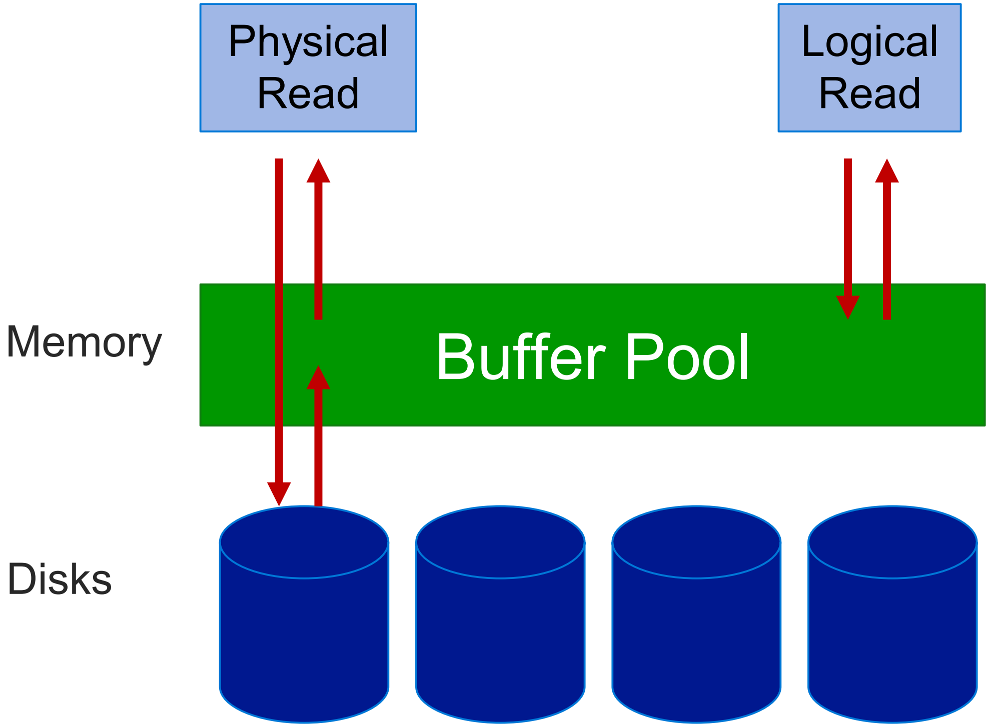
SQL Server Buffer Pool

Stores 8 kilobytes (KB) pages of data to avoid repeated disk I/O.

- Pages held in the buffer until the space is needed by something else.

Largest percentage of SQL Server memory.

- Separate buffer pool nodes for each hardware NUMA node.



```
/* physical Reads & Logical Reads can be obtained with */
SET STATISTICS IO ON
```

Lock Pages in Memory

Special operating system API for memory allocations.

Memory allocated through this API cannot be paged out by the operating system.

Needed to support large page allocations.

Configured by granting the Lock pages in memory security privilege to the SQL Server service account.

Shrinking the Buffer Pool

One Lazy Writer thread per hardware NUMA node

The Lazy Writer sweeps over the Buffer Pool when there is memory pressure to avoid SQL Server being paged

When the Lazy Writer searches for eligible buffers

- If the buffer is dirty, an asynchronous write (lazy write) is posted so that the buffer can be later freed.
- If the buffer is not dirty, it is freed.

Monitor Lazy Writer with sys.dm_os_memory_cache_clock_hands

Buffer Pool Extensions

Use non-volatile drives (solid-state drives (SSD)) to extend buffer pool.

No benefit to performance if there is enough memory in the server (adding memory is preferable, if possible).

Best to use SSD storage sized 4x-10x times of RAM size.

```
ALTER SERVER CONFIGURATION
SET BUFFER POOL EXTENSION ON
(FILENAME = 'os_file_path_and_name', SIZE = {SIZE KB/MB/GB})
```

Buffer Pool Parallel Scans

New feature in SQL Server 2022

Operations that cause a buffer pool scan.

- Database startup
- Database shutdown or restart
- AG failover
- Database removal (drop)
- File removal from a database
- Full or differential database backup

Buffer Pool Parallel Scans improves the performance of buffer pool scan operations on large-memory machines by utilizing multiple CPU cores.

There will be one task per 8 million buffers (64 GB) where a serial scan will still be used if there are less than 8 million buffers.

Questions?

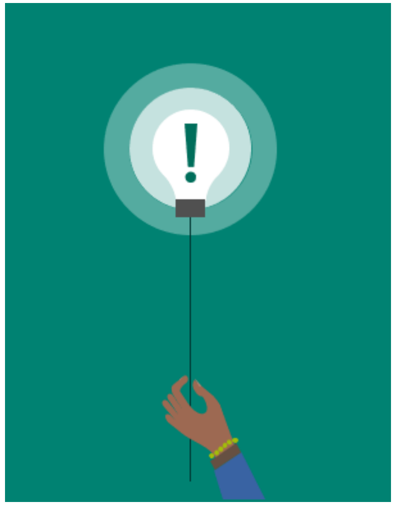
Lesson 5: Troubleshooting SQL Server Memory

Objectives

After completing this learning, you will be able to:

Understand how to monitor SQL Server memory

Identify tools for monitoring and troubleshooting memory issues



Tools to monitor SQL Server memory usage

- PERFORMANCE MONITOR
- DBCC MEMORYSTATUS
- DYNAMIC MANAGEMENT VIEWS
- RING BUFFERS WITH SYSTEM HEALTH DATA (STATUS AND OOM)
- SQL SERVER ERRORLOG (OOM WITH DBCC MEMORYSTATUS DUMP)

Performance Monitor Counters

Buffer Manager\Buffer cache hit ratio

- Must be as close to 100% as possible

Buffer Manager\Page life expectancy

- A drop of more than 30% should be investigated

Buffer Manager\Page lookups/sec

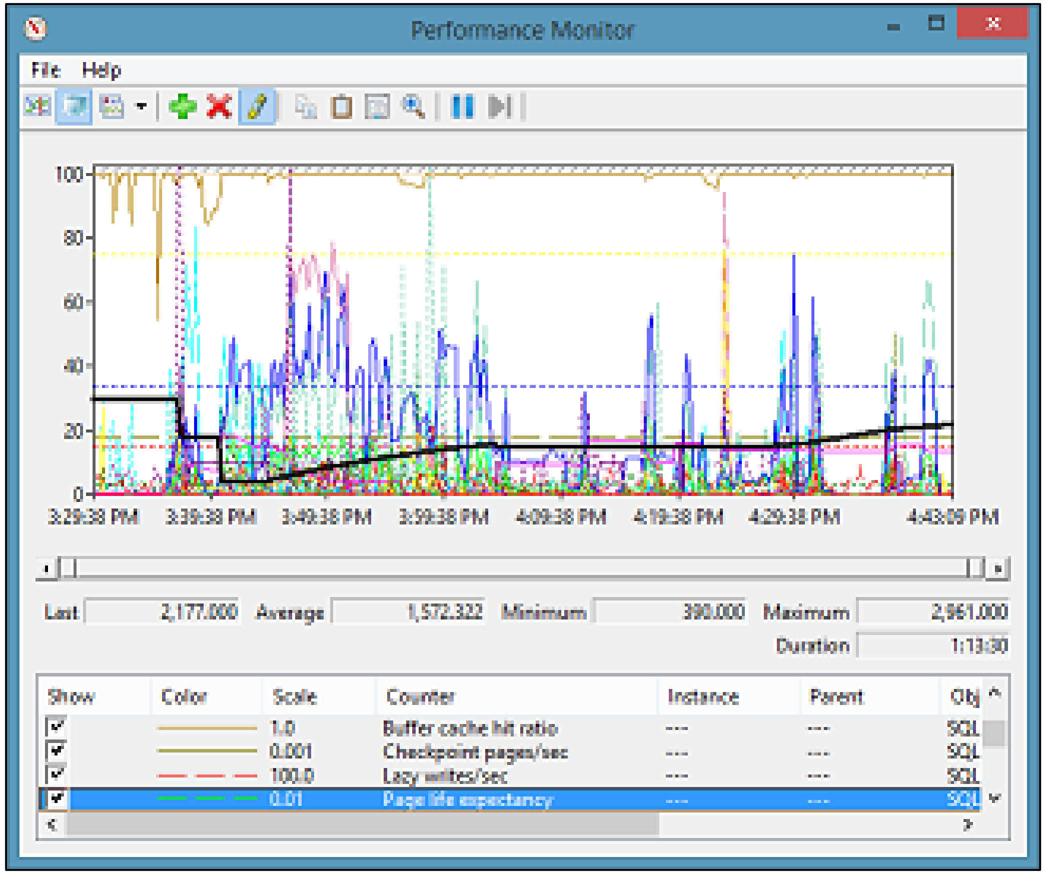
Buffer Manager\Page reads/sec

Plan Cache\Cache Hit Ratio

- Must be higher than 95%

Granted Workspace Memory (KB)

- Automatically calculated based on system configuration, monitor for high usage



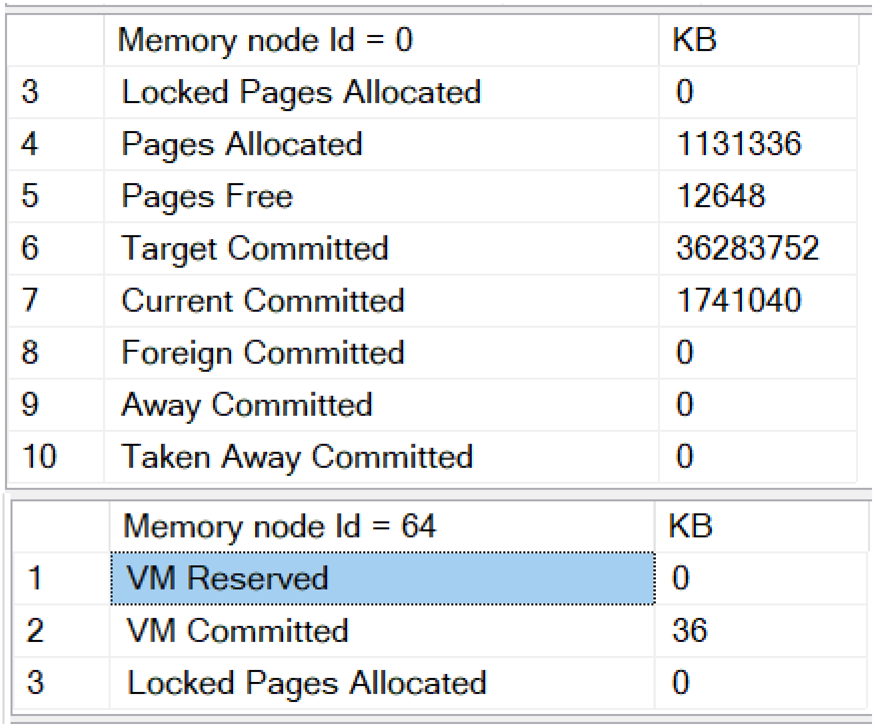
Last 2,177.000 Average 1,572.322 Minimum 390.000 Maximum 2,961.000

Duration 1:13:30

Show	Color	Scale	Counter	Instance	Parent	Obj
☑	🟨	1.0	Buffer cache hit ratio	---	---	SQL
☑	🟫	0.001	Checkpoint pages/sec	---	---	SQL
☑	🟥	100.0	Lazy writes/sec	---	---	SQL
☑	🟩	0.01	Page life expectancy	---	---	SQL

DBCC MEMORYSTATUS

- Memory Manager
- Summary of memory usage
- Aggregate memory
- Buffer distribution
- Buffer pool details
- Procedure cache
- Global memory objects
- Query memory objects
- Optimization
- Memory brokers



	Memory node Id = 0	KB
3	Locked Pages Allocated	0
4	Pages Allocated	1131336
5	Pages Free	12648
6	Target Committed	36283752
7	Current Committed	1741040
8	Foreign Committed	0
9	Away Committed	0
10	Taken Away Committed	0

	Memory node Id = 64	KB
1	VM Reserved	0
2	VM Committed	36
3	Locked Pages Allocated	0

Memory Specific DMVs

DMV Name	Purpose
sys.dm_os_process_memory	Most memory allocations that are attributed to the SQL Server
sys.dm_os_memory_brokers	Memory brokers fairly distribute memory allocations between various components within SQL Server, based on current and projected usage.
sys.dm_os_loaded_modules	Returns a row for each module loaded into the server address space
sys.dm_os_memory_objects	Returns the set of all memory clerks that are currently active in the instance of SQL Server
sys.dm_os_memory_clerks	Returns memory objects that are currently allocated by SQL Server.

Ring Buffers with System Health data (status and OOM)

Microsoft has not officially documented the sys.dm_os_ring_buffers DMV, however you can query it for many different types of records.

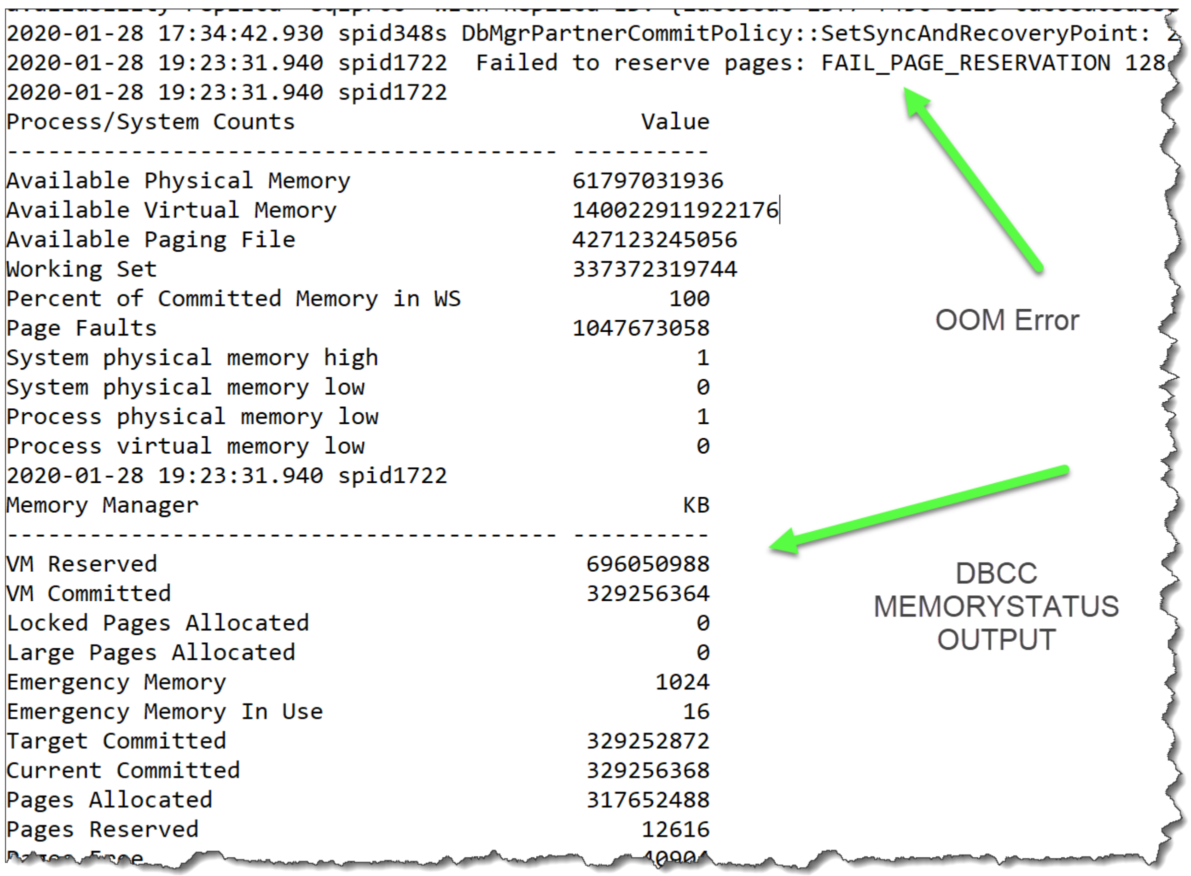
Examples of using the sys.dm_os_ring_buffers DMV to examine:

- Current memory pressure as reported by system health.
- If there have been any Out of Memory (OOM) reports.

```
SELECT CAST(record AS XML)
FROM   sys.dm_os_ring_buffers
WHERE  ring_buffer_type = 'RING_BUFFER_RESOURCE_MONITOR’;

SELECT record
FROM   sys.dm_os_ring_buffers
WHERE  ring_buffer_type = 'RING_BUFFER_OOM';
```

SQL Error Log



```
2020-01-28 17:34:42.930 spid348s DbMgrPartnerCommitPolicy::SetSyncAndRecoveryPoint: 2
2020-01-28 19:23:31.940 spid1722  Failed to reserve pages: FAIL_PAGE_RESERVATION 128
2020-01-28 19:23:31.940 spid1722
Process/System Counts                              Value
---------------------------------------- ----------
Available Physical Memory                61797031936
Available Virtual Memory                 140022911922176
Available Paging File                    427123245056
Working Set                              337372319744
Percent of Committed Memory in WS               100
Page Faults                              1047673058
System physical memory high                       1
System physical memory low                        0
Process physical memory low                       1
Process virtual memory low                        0
2020-01-28 19:23:31.940 spid1722
Memory Manager                                     KB
---------------------------------------- ----------
VM Reserved                               696050988
VM Committed                              329256364
Locked Pages Allocated                            0
Large Pages Allocated                             0
Emergency Memory                               1024
Emergency Memory In Use                          16
Target Committed                          329252872
Current Committed                         329256368
Pages Allocated                           317652488
Pages Reserved                                12616
Pages Free                                    40904
```

OOM Error

DBCC MEMORYSTATUS OUTPUT

Monitoring SQL Server Memory

Questions?

Knowledge Check

List three features that can be used to monitor memory usage?

Which DMV allows you to view memory pressure?

Which memory component is responsible for allocating memory?

Microsoft