Understanting Memory Pressure

Pedro A. Lopes
MS Premier Field Engineering

Our Sponsors:

Microsoft

Rumos
Formação Profissional

What Are Your Professional Development Goals?

I want to take the path from DBA to Data Analytics Guru

I want to upgrade my skills

I want to give my career a competitive edge

I want to expand my network in the business analytics industry

Sound familiar? Get a head start and join us today at:

PASS BUSINESS ANALYTICS CONFERENCE

APRIL 10-12 2013
CHICAGO, IL

www.passbaconference.com

#passbac

Enjoy $150 off registration: use code CHM2D

PASS SQL saturday

#188 | PORTUGAL 2013

Upcoming SQL Server events:

XXXIII Encontro da Comunidade SQLPort
Data Evento: 23 Abril 2013 - 18:30
Local do Evento: Auditório Microsoft, Parque das Nações, **Lisboa**

18:30 - Abertura e recepção.
19:10 - "Analyzing Twitter Data" - Niko Neugebauer (SQL Server MVP, Community Evangelist – PASS)
20:15 - Coffee break
20:30 - "First Approach to SQL Server Analysis Services" - João Fialho (Consultor BI Independente)
21:30 - Sorteio de prémios

XXXIV Encontro da Comunidade SQLPort
Data Evento: 7 Maio 2013 - 19:00
Local do Evento: **Porto**

18:30 - Abertura e recepção.
19:00 - «Apresentação para Developers» - para definir
20:15 - Coffee break
20:30 - «Apresentação para definir» - para definir
21:30 - Sorteio de prémios

Volunteers:

- They spend their FREE time to give you this event. (2 months per person)
- Because they are crazy.
- Because they want YOU
  to learn from the BEST IN THE WORLD.
- If you see a guy with “STAFF” on their back – buy them a beer, they deserve it.

Paulo Matos:

Paulo Borges:

João Fialho:

Bruno Basto:

Pedro A. Lopes

- SQL PFE with Microsoft since 2010
- Mainly into performance tuning, regularly deliver SQL RAPs and PTO Clinics.
- Blogs about all things SQL @ http://blogs.msdn.com/b/blogdoezequiel

Microsoft CERTIFIED IT Professional
Database Administrator 2008
Database Developer 2008

Microsoft CERTIFIED Trainer

Session Outline

- Some Memory Concepts
- Types of Memory Pressure
- Understanding memory pressure response mechanics
- Detecting External Memory Pressure
- Detecting Internal Memory Pressure

SQL Server Bottlenecks

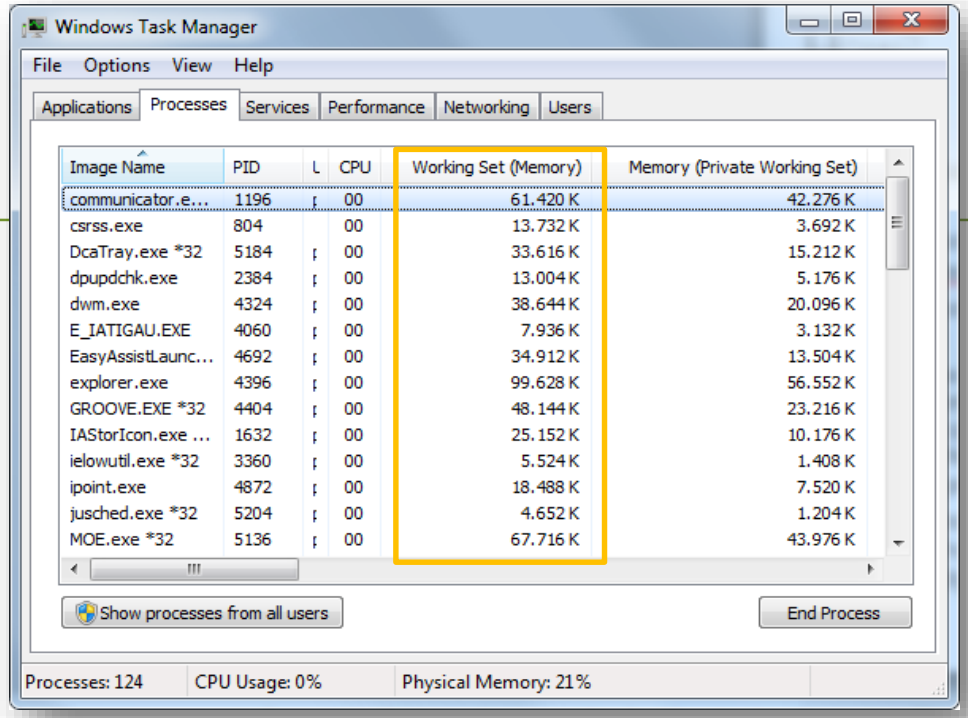
1. CPU
2. Memory / RAM
3. Disk IO / Storage Subsystem
4. Network IO
5. Locking and Blocking
6. TempDB

SOME MEMORY CONCEPTS

SQL Server Memory Models

Conventional	AWE	Large Pages
• Default mem model • "Using conventional memory in the memory manager"	• Enterprise Edition • Lock Page In Memory • Standard Edition with TF 845 enabled • “Using locked pages for buffer pool” OR • "Address Windowing Extensions enabled."	• Enterprise edition • Large page support • Lock Page In Memory • TF 834 enabled for BP use (KB920093*) • “Using large pages for buffer pool”

Memory concepts

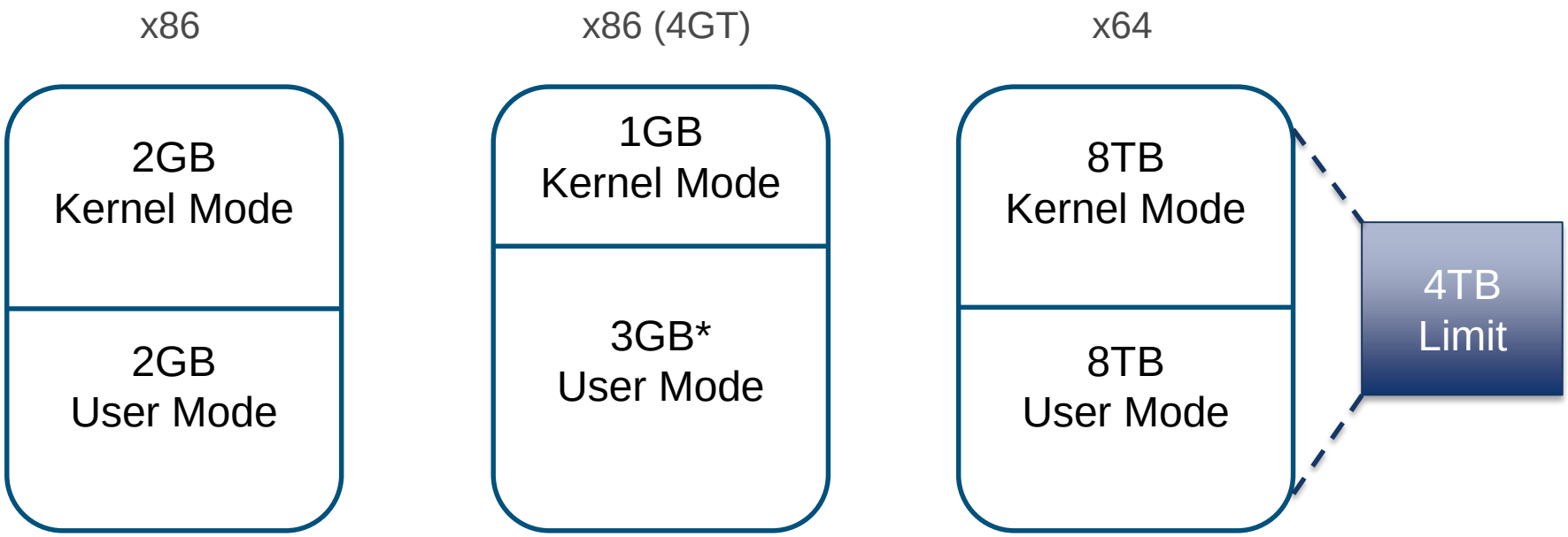


External to SQL

- Virtual Memory
  - Memory that is referenced by Windows and apps.
  - It is not the page file.
- Physical Memory
  - RAM.
- VAS
  - Set of virtual memory addresses that a process can use.
  - For each process can be smaller or larger than the total physical memory available.
- Process Working Set
  - Committed (used) memory of a process that resides in physical RAM.
  - Subset of the VAS of a process that resides in physical memory.
  - Can be trimmed under memory pressure.

Timeout – VAS?

- The Virtual Address Space for a process is the set of virtual memory addresses that it can use.



* If **IMAGE_FILE_LARGE_ADDRESS_AWARE** flag.

Timeout – How does it translate?

- VAS of each process can be smaller or larger than the total physical memory available.
- The subset of the VAS of a process that resides in physical memory is known as the *working set*.
- The total amount of VAS available to a process is limited by physical memory *and* the free space on the page file.

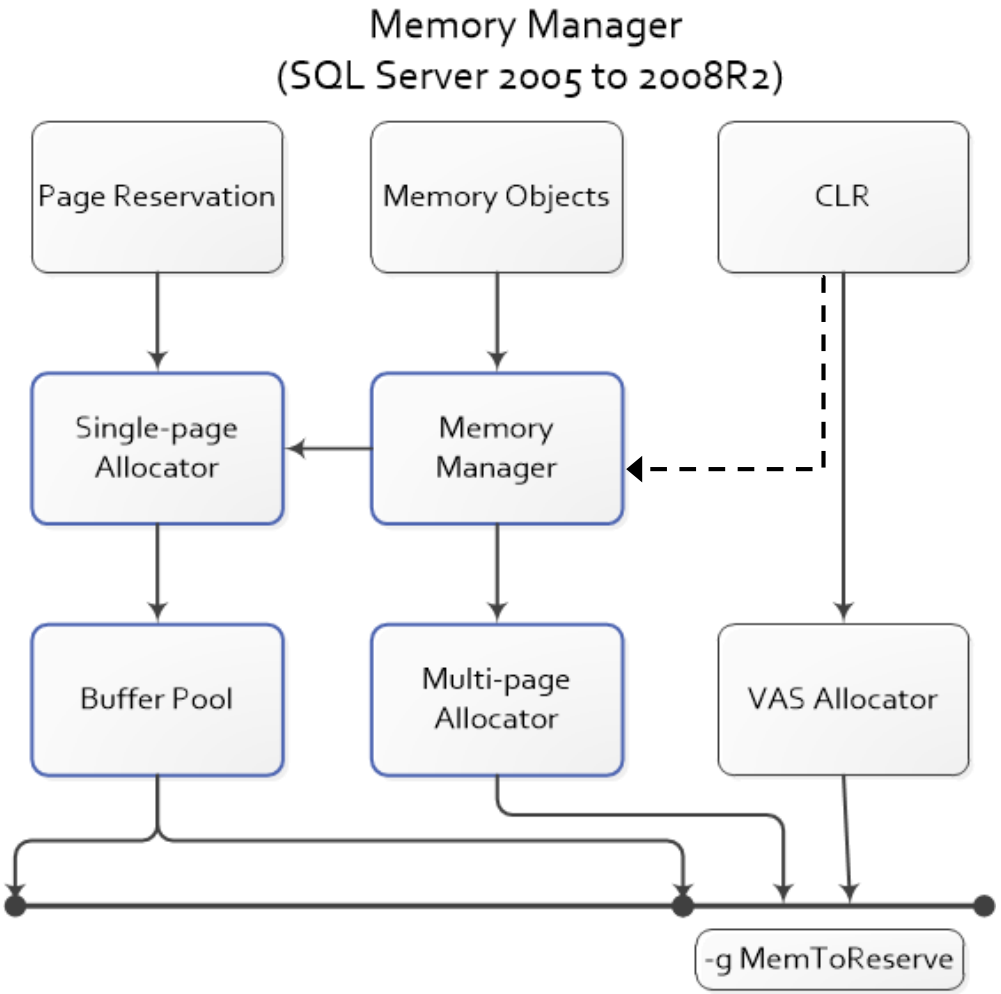
Up to SQL Server 2008 R2

Internal to SQL

- SQL Server Buffer Pool (BPool)
  - Most SQL Server memory allocated using SPAlloc (leverages Bpool).
  - So not just database pages but anything that allocates a single page.
  - Set by “max server mem” parameter.
- SQL Server (ReservedMem)
  - Memory outside BPool.
  - MPAlloc, Thread stacks, CLR, Linked Servers*, sp_OA* OLE and xProcs*
    *direct Windows allocations - DWA.
  - On x32, set automatically based on
    Stack size * ***max worker threads*** + ***-g*** startup parameter
    - Of this, only the “-g” is used for anything but the thread management.
    - Does not apply for x64 versions of SQL Server.
  - Not limited by “max server mem” parameter.

* Sometimes improperly referred as MTLeave on x64.

Up to SQL Server 2008 R2

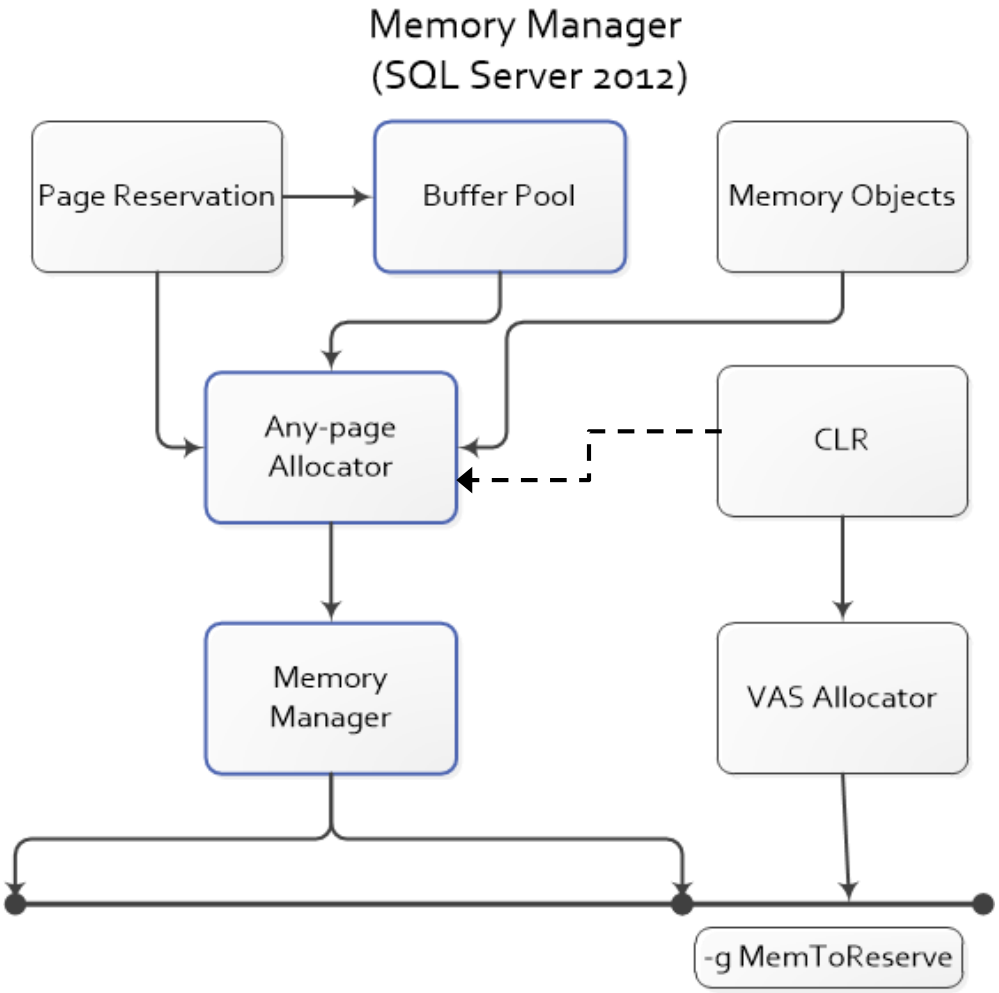


Changes in SQL Server 2012

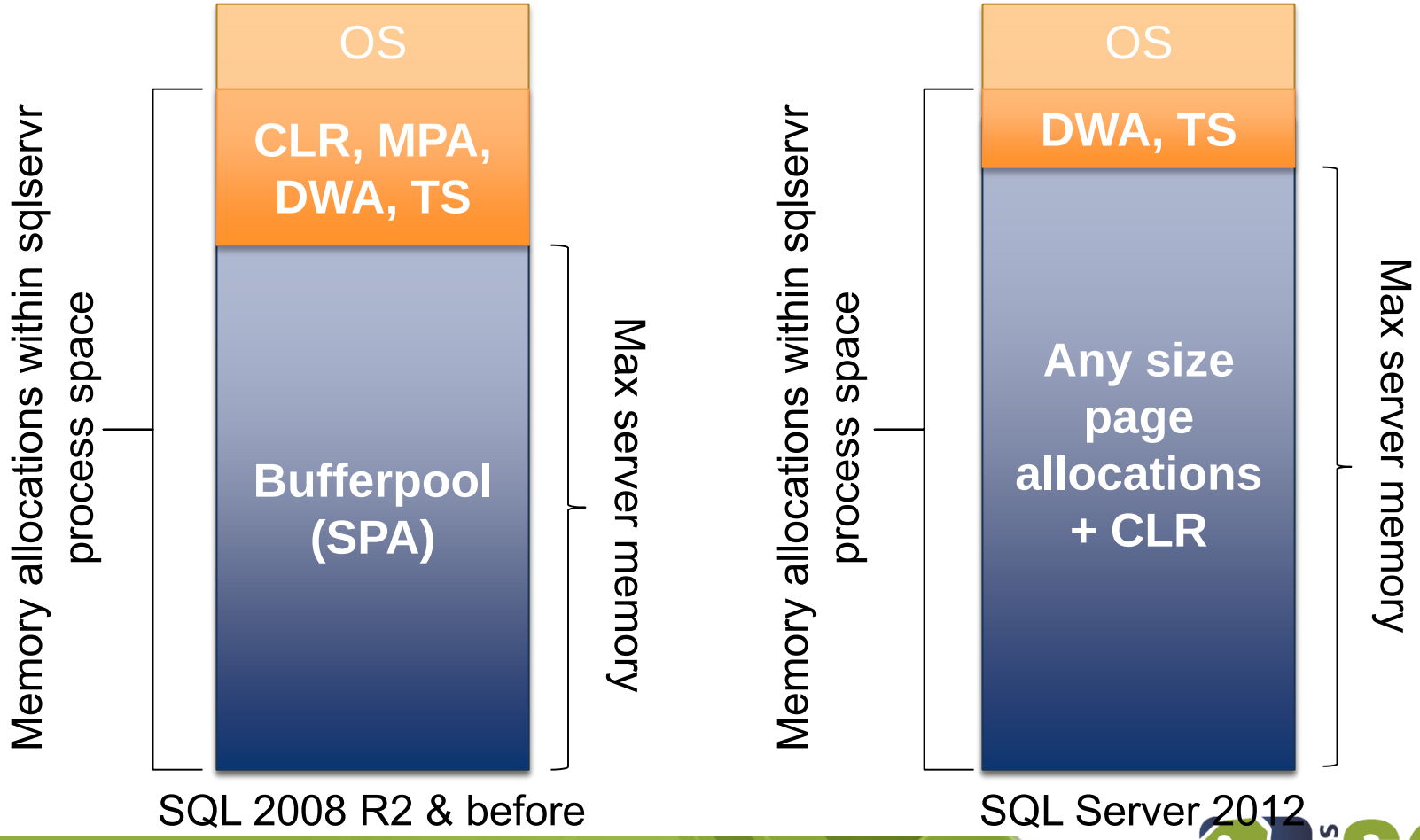
Internal to SQL

- SQL Server (Any-size page Allocator)
  - AP Allocator also allocates pages for Buffer Pool and CLR.
  - Set by “max server mem” parameter.
  - Still, most of SQL Server memory allocated is allocated by the Bpool.
  - Bpool now just storage for database pages.
- SQL Server (ReservedMem)
  - Memory outside the AP Allocator.
  - Thread stacks, Linked Servers*, sp_OA* OLE and xProcs*
    **direct Windows allocations - DWA*.
  - On x32 the algorithm remains, such as the default for x64.
  - Not limited by “max server mem” parameter.

Changes in SQL Server 2012



Memory Concepts - Summary



DWA – Direct Windows Allocations; TS – Thread stacks

Types of Memory Pressure

MEMORY PRESSURE

External Memory Pressure

- Physical
  - Physical memory (RAM) running low.
    - This causes system to trim working sets of currently running processes which may result in overall slowdown.
    - SQL may reduce the commit target of the buffer pool and start clearing internal caches often.
- Virtual
  - Running low on space in the system page file(s).
    - This may cause the system to fail memory allocations, as it is unable to page out currently allocated memory.
    - This condition may result in the whole system responding very slowly or even bring it to a halt.

Internal Memory Pressure

- Physical
  - SQL is responding to external memory pressure (SQL Server sets lower memory usage caps).
    - Someone changed memory settings (e.g. ‘max server memory’).
    - Changes in memory distribution of internal components.
- Virtual
  - Running low on VAS due to fragmentation or running out of VAS (usually a process terminates).
  - SQL Server detects this condition and may:
    - Release reserved regions of VAS
    - Reduce buffer pool / mem manager commit target
    - Start shrinking caches.

Let’s dive for a while

Response mechanics

UNDERSTANDING MEMORY PRESSURE

Is it just memory that is affected?

NO!!

- Memory pressure situations might trigger:
  - Extra I/Os
  - Higher recompile ratio
  - Longer queries (if they have to wait for grants)
  - Extra CPU
  - Other unnecessary activities
- SQL Server has an entire framework dedicated to the detection of memory pressure, of which stand out the…

Ring Buffers – the “mail”

- Internal message buffers from each of the engine components.
- Each ring buffer keeps a record of the last number of notifications of a certain kind:
  - RING_BUFFER_RESOURCE_MONITOR - Resource Monitor activity like was memory pressure signaled or not.
  - RING_BUFFER_SINGLE_PAGE_ALLOCATOR – Shows when the Buffer Pool single page

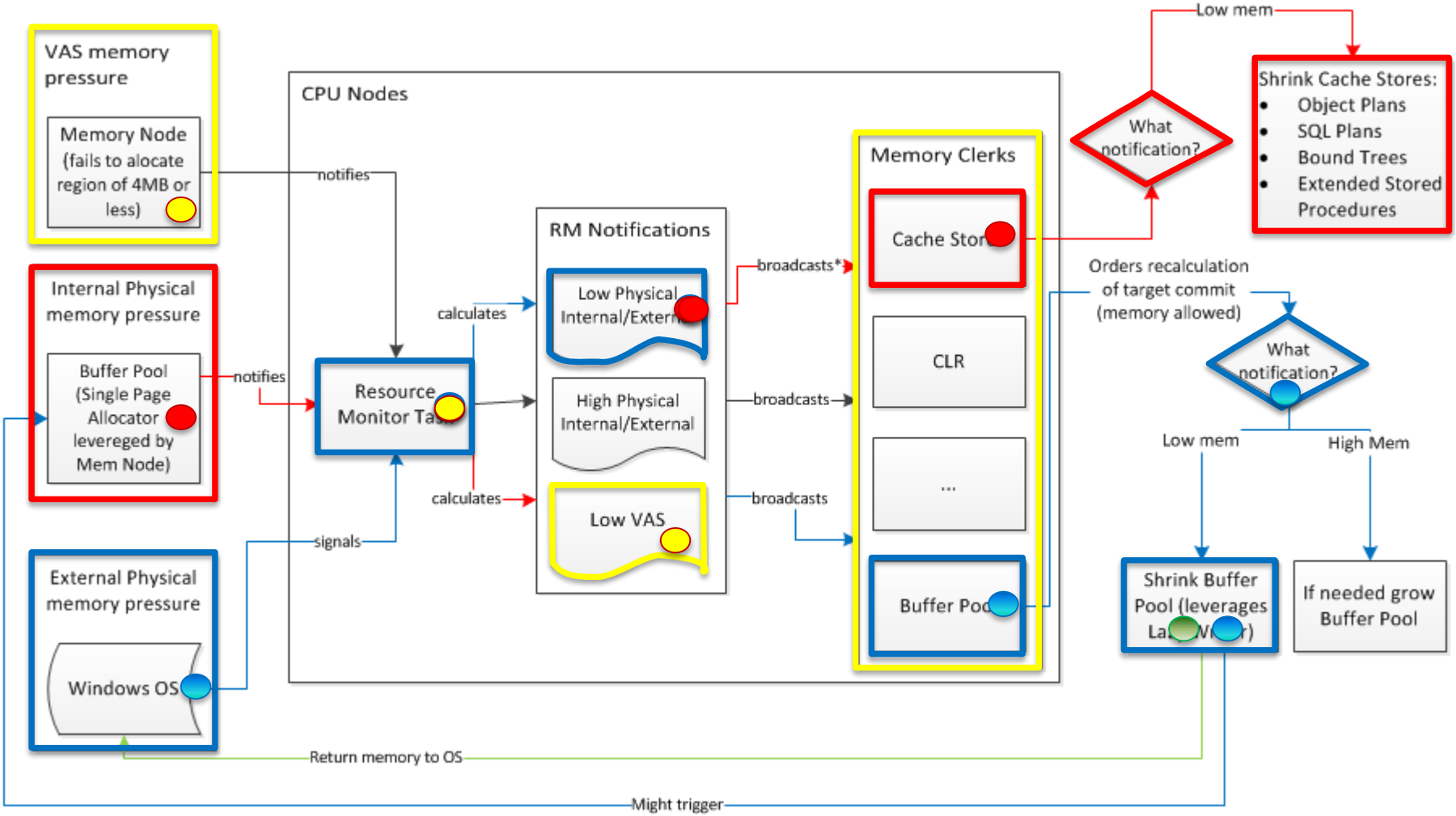
SQLOS implements a framework to enable memory pressure handling. In the heart of the framework lies **Resource Monitor task**, RM. **RM monitors state of the external and internal memory** indicators. Once one of them changes, it calculates the corresponding notification and **broadcasts it.**

  - can reconstruct the execution order from these entries.
  - RING_BUFFER_EXCEPTION - Any exceptions encountered in the server. SQL uses throw internally for errors so you can see SQL errors as well.
  - RING_BUFFER_CONNECTIVITY - Core connectivity information - useful in tracking down connection failure information
  - RING_BUFFER_CLRAPPDOMAIN - State of AppDomains loaded in the system.

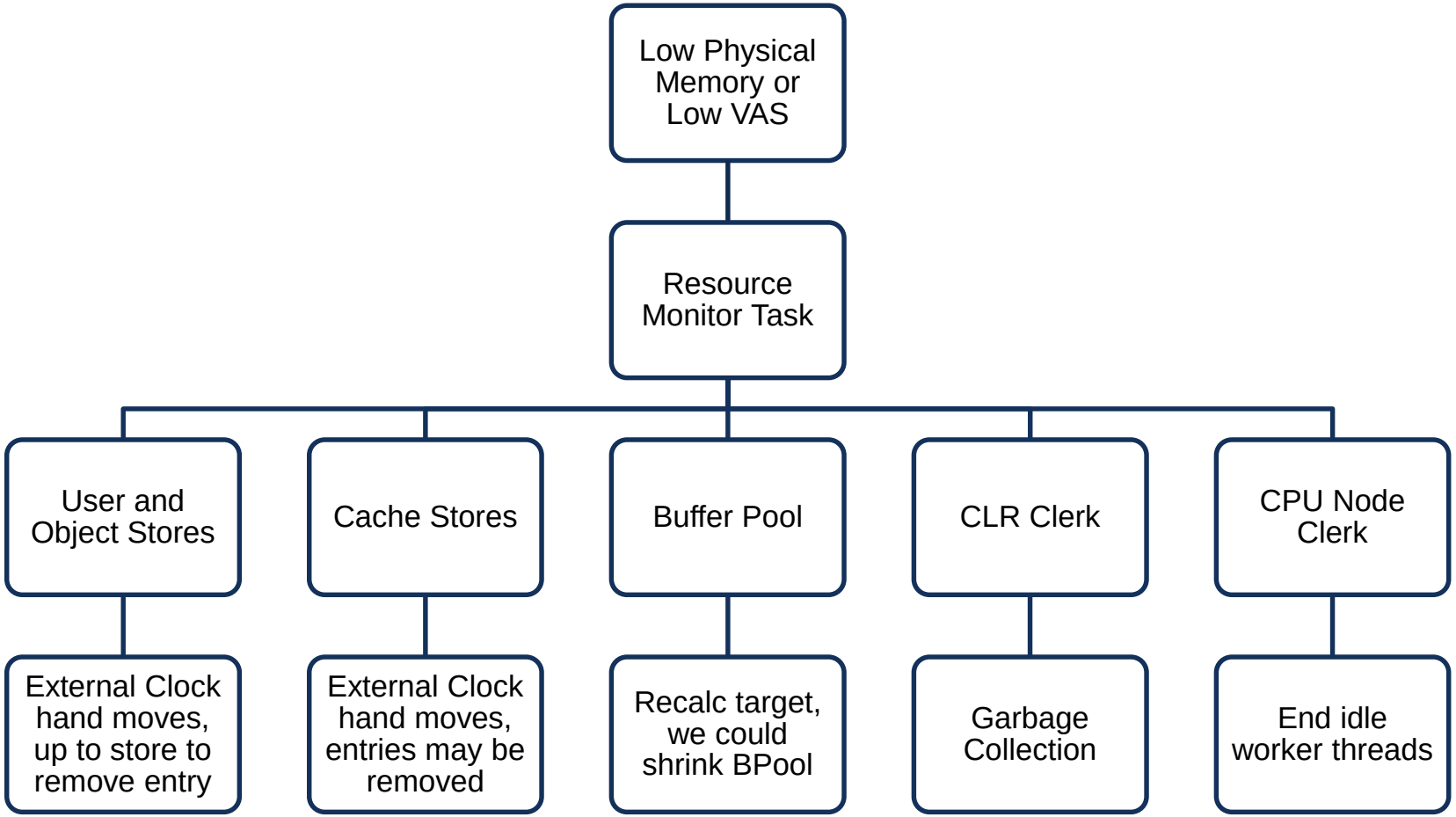
Memory Clerks – the “postman”

- A memory clerk is an accountant of memory.
- SQL Server contains many caches, pools and other memory objects.
- Most of these are wrapped by a clerk, so the various cache pool and object allocation counts are tracked.
- 76 clerks in SQL 2012, 62 in SQL 2008, 55 in SQL 2005.
  - That only means more components now rely on clerks for memory allocations.
- Memory clerks also process notifications from RM task and directs it to the appropriate destination (e.g. Buffer Pool, Cache Stores, CLR, Lock Manager, Security, etc.).

How is it related?



RM Task and Clerks in a nutshell



Coming back to the surface…

Memory

TOOLS FOR TROUBLESHOOTING

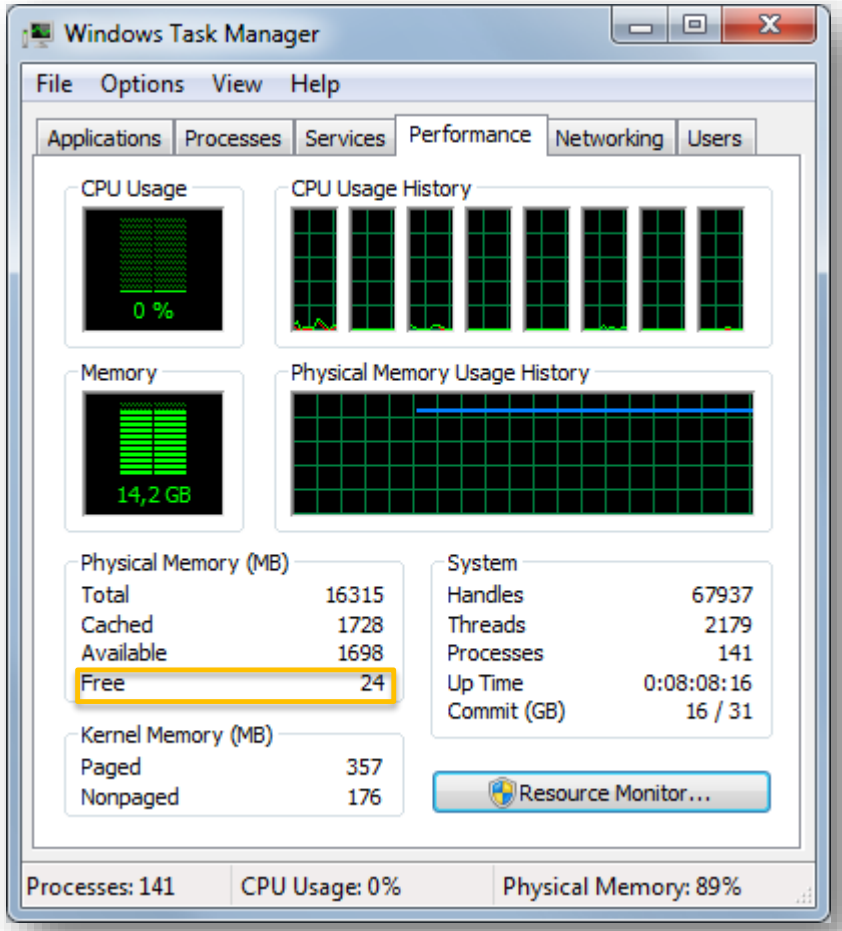
Tools for troubleshooting

- Task Manager (not valid for VMs)
- Performance counters
  - Performance monitor
  - DMV for SQL Server specific object (*sys.dm_os_performance_counters*)
- Memory related DMVs
- DBCC MEMORYSTATUS command
- Event viewer: application log, system log
- Errorlog

DETECTING EXTERNAL MEMORY PRESSURE

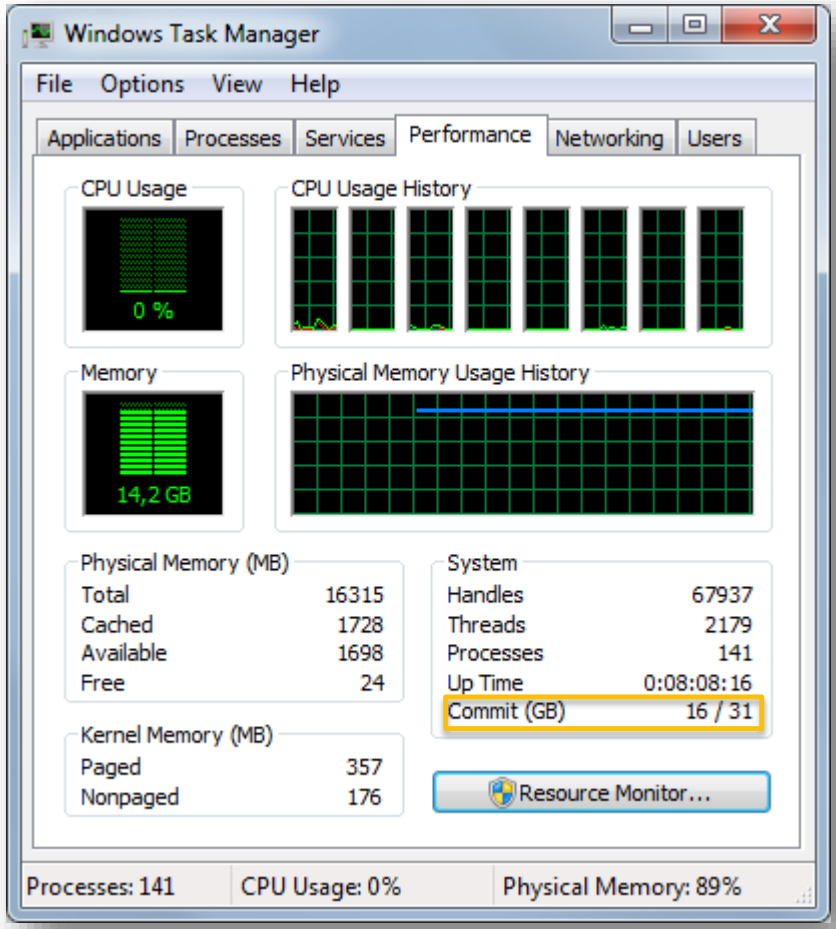
Physical - Task Manager

- Task Manager (performance view)
  - Physical Memory Available
    - < 100 MB triggers Low Memory Condition
    - With resource monitor, identify largest consumers.



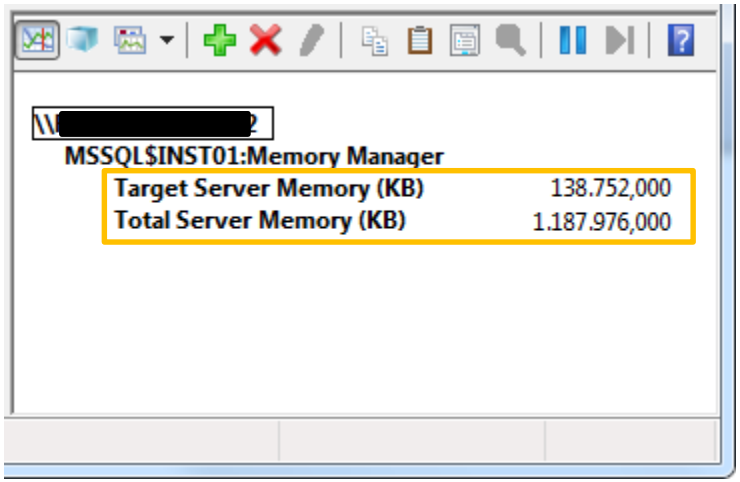
Virtual - Task Manager

- Task Manager (performance view)
  - Commit Charge (limit is the Maximum Amount of Committed Memory)
    - Is Total close to the Limit e.g. low on virtual memory?
    - Commit Charge Total indicates “potential” Page File Usage.



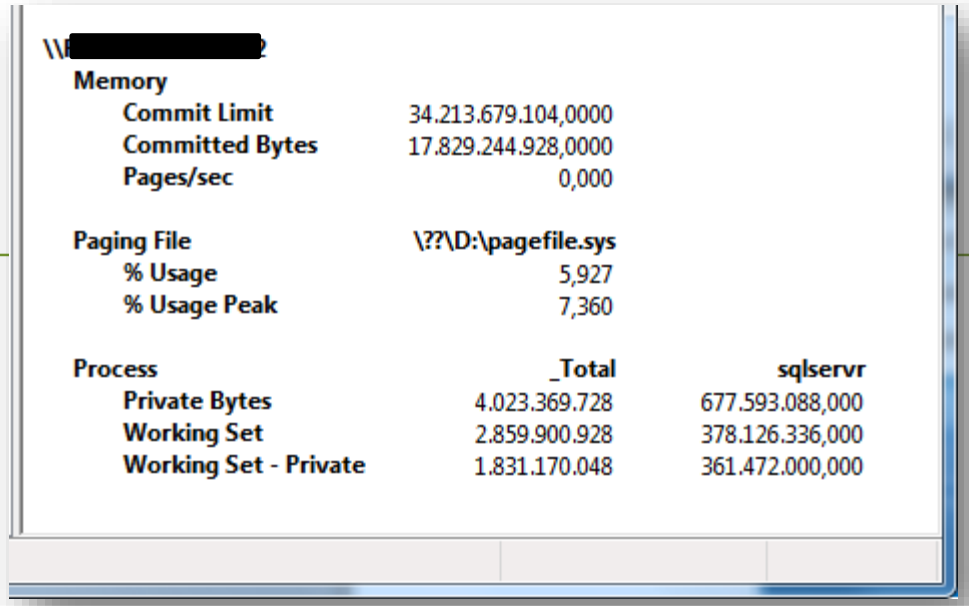
Physical – RM/Perfmon

- Resource Monitor / Performance Monitor
  - Memory Manager: Target Server Memory
  - Memory Manager: Total Server Memory



  - A lower commit value was set, indicating external physical memory pressure.

Virtual – Perfmon



\\[redacted]		
Memory		
Commit Limit	34.213.679.104,0000	
Committed Bytes	17.829.244.928,0000	
Pages/sec	0,000	
Paging File	\??\D:\pagefile.sys	
% Usage	5,927	
% Usage Peak	7,360	
Process	**_Total**	**sqlservr**
Private Bytes	4.023.369.728	677.593.088,000
Working Set	2.859.900.928	378.126.336,000
Working Set - Private	1.831.170.048	361.472.000,000

- System Monitor/Perfmon
  - Memory: Pages/sec. Page I/O to resolve references no longer in memory.
  - Memory: Free System Page Table Entries <5000 is a symptom of virtual memory constraints, namely on x86 machines.
  - Memory: Commit Limit – Memory: Commited Bytes = mem that can be commited before extending the pagefile.
  - Process: Private Bytes - Process: Working Set = Amount of process Paged Out
  - Process: Working Set (Private Bytes). Identify highest consumers of non-shareable memory.
  - Paging File: %Usage. High Value equals physical memory over commitment.
  - Paging File: %Usage Peak. Check System Event log for file growth.

Physical / Virtual - DMVs

- Useful DMVs
  - sys.dm_os_memory_clerks
  - sys.dm_os_ring_buffers
  - sys.dm_os_process_memory
  - sys.dm_os_sys_memory

New in SQL Server 2008

```
SELECT * FROM sys.dm_os_process_memory
```

Results | Messages

	physical_memory_in_use_kb	large_page_allocations_kb	locked_page_allocations_kb	total_virtual_address_space_kb	virtual_address_space_reserved_kb	virtual_address_space_committed_kb
1	289300	0	0	8589934464	23569896	660284

virtual_address_space_available_kb	page_fault_count	memory_utilization_percentage	available_commit_limit_kb	process_physical_memory_low	process_virtual_memory_low
8566364568	159801	80	16492856	0	0

Results | Messages

	total_physical_memory_kb	available_physical_memory_kb	total_page_file_kb	available_page_file_kb	system_cache_kb	kernel_paged_pool_kb	kernel_nonpaged_pool_kb
1	16707084	2227016	33411796	16627540	1851504	379328	185764

system_high_memory_signal_state	system_low_memory_signal_state	system_memory_state_desc
1	0	Available physical memory is high

Physical / Virtual - DMVs

2005
2008

- sys.dm_os_memory_clerks

```
--All memory used without BP:
SELECT SUM(single_pages_kb + multi_pages_kb +
    virtual_memory_committed_kb +
    shared_memory_committed_kb) AS Alloc_Mem_except_BP_kb
FROM sys.dm_os_memory_clerks WHERE type <> 'MEMORYCLERK_SQLBUFFERPOOL';

--All memory used by BP:
SELECT SUM(single_pages_kb + multi_pages_kb +
    virtual_memory_committed_kb +
    shared_memory_committed_kb +
    awe_allocated_kb) AS Alloc_Mem_just_BP_kb
FROM sys.dm_os_memory_clerks
WHERE type = 'MEMORYCLERK_SQLBUFFERPOOL';
```

Results | Messages

	Alloc_Mem_except_BP_kb
1	151568

	Alloc_Mem_just_BP_kb
1	277288

Physical / Virtual - DMVs

2005
2008

- sys.dm_os_memory_clerks

```
 --All memory used without BP:
SELECT SUM(pages_kb +
    virtual_memory_committed_kb +
    shared_memory_committed_kb) AS Alloc_Mem_except_BP_kb
FROM sys.dm_os_memory_clerks WHERE type <> 'MEMORYCLERK_SQLBUFFERPOOL'

--All memory used by BP:
SELECT SUM(pages_kb +
    virtual_memory_committed_kb +
    shared_memory_committed_kb +
    awe_allocated_kb) AS Alloc_Mem_just_BP_kb
FROM sys.dm_os_memory_clerks
WHERE type = 'MEMORYCLERK_SQLBUFFERPOOL'
```

100 %

Results | Messages

	Alloc_Mem_except_BP_kb
1	389136

	Alloc_Mem_just_BP_kb
1	2826952

Physical / Virtual - DMVs

2005
2008

```
WITH cte (type, allocated_kb) AS (SELECT type, SUM(single_pages_kb + multi_pages_kb +
    virtual_memory_committed_kb + shared_memory_committed_kb) AS allocated_kb
    FROM sys.dm_os_memory_clerks GROUP BY type)
SELECT mc.[type] AS clerk_type, allocated_kb,
    CAST(100.0 * allocated_kb / SUM(allocated_kb) OVER() AS DECIMAL(12, 2)) AS pct,
    SUM(single_pages_kb + multi_pages_kb) AS pages_kb,
    SUM(single_pages_kb) AS single_pages_kb,
    SUM(multi_pages_kb) AS multi_pages_kb,
    SUM(virtual_memory_committed_kb) AS virtual_memory_committed_kb,
    SUM(shared_memory_committed_kb) AS shared_memory_committed_kb,
    SUM(awe_allocated_kb) AS awe_allocated_kb,
    SUM(virtual_memory_reserved_kb) AS virtual_memory_reserved_kb,
    SUM(shared_memory_reserved_kb) AS shared_memory_reserved_kb
FROM sys.dm_os_memory_clerks mc INNER JOIN cte ON mc.type = cte.type
GROUP BY mc.type, cte.allocated_kb
ORDER BY allocated_kb DESC
```

Results | Messages

	clerk_type	allocated_kb	pct	pages_kb	sing		
1	MEMORYCLERK_SQLBUFFERPOOL	277288	62.44	424	0		
2	CACHESTORE_SQLCP	30328	6.83	30328	28		
3	MEMORYCLERK_SQLCLR	27960	6.30	10904	83		
4	MEMORYCLERK_SQLGENERAL	25792	5.81	25792	17		
5	MEMORYCLERK_SOSNODE	20752	4.67	20752	40		
6	CACHESTORE_OBJCP	11800	2.66	11800	11		
7	MEMORYCLERK_SQLSTORENG	9824	2.21	6760	22		
8	OBJECTSTORE_LOCK_MANAGER	9312	2.10	1120	1120	0	8192

- include the plans for adhoc, autoparameterized and prepared plans.
- structures produced by the engine algebrizer when binding views, constraints and defaults in the QP compilation time.

Physical / Virtual - DMVs

2012

```sql
WITH cte (type, allocated_kb) AS (SELECT type, SUM(pages_kb +
    virtual_memory_committed_kb + shared_memory_committed_kb) AS allocated_kb
    FROM sys.dm_os_memory_clerks GROUP BY type)
SELECT mc.[type] AS clerk_type, allocated_kb,
    CAST(100.0 * allocated_kb / SUM(allocated_kb) OVER() AS DECIMAL(12, 2)) AS pct,
    SUM(pages_kb) AS pages_kb,
    SUM(virtual_memory_committed_kb) AS virtual_memory_committed_kb,
    SUM(shared_memory_committed_kb) AS shared_memory_committed_kb,
    SUM(awe_allocated_kb) AS awe_allocated_kb,
    SUM(virtual_memory_reserved_kb) AS virtual_memory_reserved_kb,
    SUM(shared_memory_reserved_kb) AS shared_memory_reserved_kb
FROM sys.dm_os_memory_clerks mc
    INNER JOIN cte ON mc.type = cte.type
GROUP BY mc.type, cte.allocated_kb
ORDER BY allocated_kb DESC
```

Results | Messages

	clerk_type	allocated_kb	pct		
1	MEMORYCLERK_SQLBUFFERPOOL	2826952	87.85	2	
2	CACHESTORE_SQLCP	101192	3.14	1	
3	OBJECTSTORE_LOCK_MANAGER	81696	2.54	1	
4	CACHESTORE_PHDR	58528	1.82	5	
5	MEMORYCLERK_SOSNODE	32608	1.01	3	
6	MEMORYCLERK_SQLCLR	29976	0.93	2	
7	MEMORYCLERK_SOSMEMMANAGER	21504	0.67	0	
8	MEMORYCLERK_SQLGENERAL	11096	0.34	11096	0

- include the plans for adhoc, autoparameterized and prepared plans.
- structures produced by the engine algebrizer when binding views, constraints and defaults in the QP compilation time.

Physical - DMVs

- sys.dm_os_ring_buffers
- 'RING_BUFFER_RESOURCE_MONITOR'
  - Shows a set of last notification RM broadcasted.
  - BPool clerk leverages RM notifications for external physical memory pressure.

```
1  -- Checks last Resource Monitor task message
2  --
3  SELECT --record.value('(/Record/ResourceMonitor/Notification)[1]', 'varchar(max)') AS [Type],
4      record, DATEADD(ms, x.[timestamp] - si.ms_ticks, GETDATE()) AS Event_Time
5  FROM (SELECT MAX([TIMESTAMP]) AS [TIMESTAMP], CONVERT(xml, record) AS record
6              FROM sys.dm_os_ring_buffers
7              WHERE ring_buffer_type = N'RING_BUFFER_RESOURCE_MONITOR'
8              GROUP BY record) AS x
9  CROSS APPLY sys.dm_os_sys_info si
```

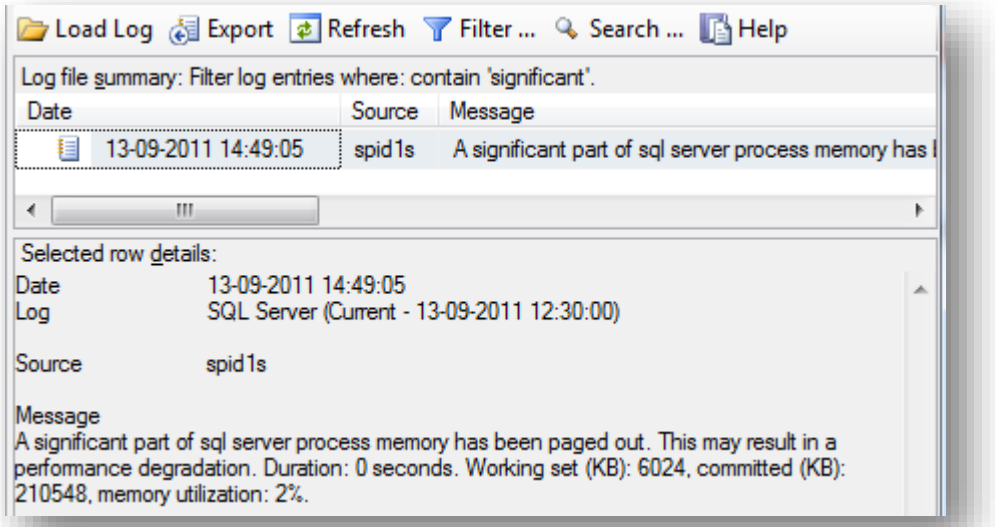
100 %

Results | Messages

		Event_Time
1	MONITOR" time="251818732"><ResourceMonitor><Notification>RESOURCE_MEMPHYSICAL_HIGH</Notification>...	2012-11-24 18:18:13.833
2	MONITOR" time="253036539"><ResourceMonitor><Notification>RESOURCE_MEMPHYSICAL_LOW</Notification>...	2012-11-24 18:38:31.640

Physical - Logged memory errors

- In SQL 2005 SP2 and later, if the working set of a SQL Server process is forcibly paged out by OS, this type of message is logged in the ERRORLOG:

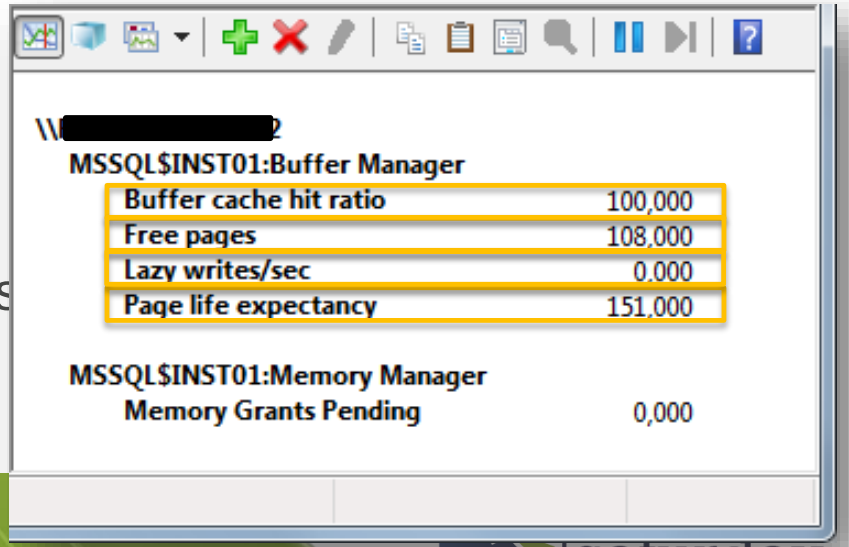


  - It's generated when RAM used by SQL Server (Private Working Set) drops under 50% of the entire virtual memory used (Committed Bytes or Virtual Bytes).
  - Preventable if LPIM is granted.

DETECTING INTERNAL MEMORY PRESSURE

Perfmon

- System Monitor/Perfmon
  - Buffer Manager: Buffer Cache Hit Ratio > 90 % for OLTP
  - Buffer Manager: Lazy writes/sec – Possibly 0 (non-NUMA)
  - Buffer Node: Lazy writes/sec – Possibly 0 (NUMA)
    - Buffer Manager: Checkpoint Pages/sec (correlate)
  - Buffer Manager: Free Pages > 640* (5MB)
    - Memory Manager: Free Memory (KB) in SQL Server 2012
  - Buffer Manager: Page Life Expectancy (sec) > 300*
  - Memory Manager: Memory Grants Pending – Possibly 0 for OLTP



* Already a **very** low threshold

Memory related DMVs

- Useful DMVs
  - `sys.dm_os_ring_buffers`
  - `sys.dm_os_memory_clerks`
  - `sys.dm_os_memory_cache_counters`

Memory related DMVs

- `sys.dm_os_ring_buffers`
- `'RING_BUFFER_SINGLE_PAGE_ALLOCATOR'`
  - Shows when BPool single-page allocator turns on/off internal memory pressure bit.
  - Remember BPool ignores RM notifications of internal physical memory pressure.
  - Refer to Process/System Counts section in DBCC MEMORYSTATUS

```
SQLQuery1.sql - (l...OPE\pelopes (52))*
SELECT [timestamp], record
FROM sys.dm_os_ring_buffers
WHERE ring_buffer_type='RING_BUFFER_SINGLE_PAGE_ALLOCATOR'
```

Memory related DMVs

2005
2008

- `sys.dm_os_memory_clerks`
- At startup, BPool is configured to be SQLOS's single page allocator.
- From that point on all dynamic single page allocations (8KB) are provided by the Bpool, and typically it consumes the most memory.

SQLQuery1.sql - [redacted]2))*

```
SELECT TOP 10 type, SUM(single_pages_kb) AS [SPA Mem, Kb]
FROM sys.dm_os_memory_clerks
GROUP BY type
ORDER BY SUM(single_pages_kb) DESC
```

Results | Messages

	type	SPA Mem, Kb
1	CACHESTORE_SQLCP	55056
2	CACHESTORE_PHDR	27352
3	OBJECTSTORE_LOCK_MANAGER	13600
4	USERSTORE_DBMETADATA	7592
5	CACHESTORE_OBJCP	6936
6	MEMORYCLERK_SOSNODE	5232
7	USERSTORE_SCHEMAMGR	4600
8	CACHESTORE_SYSTEMROWSET	3504
9	MEMORYCLERK_SQLSTORENG	2144
10	MEMORYCLERK_SQLGENERAL	2040

- include the plans for adhoc, autoparameterized and prepared plans.
- structures produced by the engine algebrizer when binding views, constraints and defaults in the QP compilation time.
- include plans for stored procedures, functions, and triggers.

Memory related DMVs

2005
2008

- Other components can be consuming memory other than BP:
  - If the needed allocation is > 8KB, these components use memory outside the BPool through the MPA interface, which allocates sets of contiguous 8KB pages.
  - If a significant amount of memory is allocated through the MPA (~100-~200 MB or more), further investigation is warranted.
  - Not limited by MaxMem.

```
-- View memory allocated thru MPA
SELECT type, sum(multi_pages_kb) AS multi_pages_kb
FROM sys.dm_os_memory_clerks
WHERE multi_pages_kb <> 0
```

Results | Messages

	type	multi_pages_kb
1	CACHESTORE_OBJCP	10208
2	CACHESTORE_PHDR	112
3	CACHESTORE_SQLCP	26424
4	CACHESTORE_STACKFRAMES	40
5	MEMORYCLERK_SNI	80
6	MEMORYCLERK_SOSNODE	34800
7	MEMORYCLERK_SQLBUFFERPOOL	2456
8	MEMORYCLERK_SQLGENERAL	7448
9	MEMORYCLERK_SQLOPTIMIZER	928
10	MEMORYCLERK_SQLSERVICEBROKER	544
11	MEMORYCLERK_SQLSTORENG	5088
12	MEMORYCLERK_XE	168
13	OBJECTSTORE_LBSS	800
14	OBJECTSTORE_SNI_PACKET	472
15	USERSTORE_SCHEMAMGR	1904
16	USERSTORE_TOKENPERM	3064

Detecting Internal Memory Pressure

2005
2008

```
--Amount of memory per cache store
SELECT [type], name,
     SUM(single_pages_kb + multi_pages_kb) AS pages_kb,
     SUM(single_pages_in_use_kb + multi_pages_in_use_kb) AS pages_in_use_kb,
     SUM(entries_count) AS entries_count,
     SUM(entries_in_use_count) AS entries_in_use_count
FROM sys.dm_os_memory_cache_counters
WHERE type LIKE 'CACHE%'
GROUP BY name, type
ORDER BY pages_kb DESC;
```

Results | Messages

	type	name				entries_in_use_count
1	CACHESTORE_OBJCP	Object Plans				0
2	CACHESTORE_SQLCP	SQL Plans				3
3	CACHESTORE_PHDR	Bound Trees				0
4	CACHESTORE_SYSTEM...	SystemRowsetStore				0
5	CACHESTORE_BROKER...	Broker dormant rowsets				0
6	CACHESTORE_CONVPRI	ConversationPriorityCache				0
7	CACHESTORE_BROKER...	Service broker mapping table				0
8	CACHESTORE_FULLTEX...	FTSTOPLIST_CACHESTORE				0
9	CACHESTORE_XPROC	Extended Stored Procedures				0
10	CACHESTORE_TEMPTAB...	Temporary Tables & Table Var...	24	0	15	0
11	CACHESTORE_VIEWDEF...	View Definition Cache	16	0	0	0

- include plans for stored procedures, functions, and triggers.
- include the plans for adhoc, autoparameterized and prepared plans.
- structures produced by the engine algebrizer when binding views, constraints and defaults in the QP compilation time.

Detecting Internal Memory Pressure

2012

```sql
--Amount of memory per cache store
SELECT [type], name,
     SUM(pages_kb) AS pages_kb,
     SUM(pages_in_use_kb) AS pages_in_use_kb,
     SUM(entries_count) AS entries_count,
     SUM(entries_in_use_count) AS entries_in_use_count
FROM sys.dm_os_memory_cache_counters
WHERE type LIKE 'CACHE%'
GROUP BY name, type
ORDER BY pages_kb DESC;
```

Results | Messages

type	name				entries_in
CACHESTORE_SQLCP	SQL Plans				4
CACHESTORE_PHDR	Bound Trees				0
CACHESTORE_SYSTEMROW...	SystemRowsetStore				0
CACHESTORE_SEHOBTCOL...	SESHAREDCOLMETADATACA...				0
CACHESTORE_CONVPRI	ConversationPriorityCache				0
CACHESTORE_BROKERTBL...	Broker dormant rowsets				0
CACHESTORE_COLUMNSTO...	Column store object pool	64	0	0	0
CACHESTORE_FULLTEXTST...	FTSTOPLIST_CACHESTORE	56	0	0	0
CACHESTORE_XPROC	Extended Stored Procedures	32	0	2	0
CACHESTORE_STACKFRAM...	SOS_StackFramesStore	16	8	1	1

- include the plans for adhoc, autoparameterized and prepared plans.
- structures produced by the engine algebrizer when binding views, constraints and defaults in the QP compilation time.

DBCC MEMORYSTATUS

- **Committed** = number of 8KB pages* currently committed.
- **Target** = number of 8KB pages* the BPool will have before paging occurs (never larger than MaxServerMem).

2005 Output

Buffer Counts	Buffers
Committed	1064
Target	17551
Hashed	345
Stolen Potential	121857
External Reservation	645
Min Free	64
Visible	17551
Available Paging File	451997

Buffer Distribution	Buffers
Stolen	553
Free	103
Cached	161
Database (clean)	1353
Database (dirty)	38
I/O	0
Latched	0

2008 Output

Buffer Pool	Value
Committed	39664
Target	384000
Database	14582
Dirty	1551
In IO	0
Latched	0
Free	18947
Stolen	6135
Reserved	0
Visible	384000
Stolen Potential	358665
Limiting Factor	17
Last OOM Factor	0
Page Life Expectancy	1267

2012 Output

Memory Manager	KB
VM Reserved	29551632
VM Committed	745624
Locked Pages Allocated	0
Large Pages Allocated	0
Emergency Memory	1024
Emergency Memory In Use	16
Target Committed	7786376
Current Committed	745624
Pages Allocated	203960
Pages Reserved	0
Pages Free	345136
Pages In Use	336504
Page Alloc Potential	8087616
NUMA Growth Phase	0
Last OOM Factor	6
Last OS Error	0

* Expressed in KB on 2012

DBCC MEMORYSTATUS

- **Target** value will change depending on memory low/high notifications from Resource Monitor task:
  - If the **Target** count is greater than the **Committed** value, the BP* is growing.
  - If the **Target** count is less than the **Committed** value, the BP* is shrinking.

2005 Output

Buffer Counts	Buffers
Committed	1064
Target	17551
Hashed	345
Stolen Potential	121857
External Reservation	645
Min Free	64
Visible	17551
Available Paging File	451997

Buffer Distribution	Buffers
Stolen	553
Free	103
Cached	161
Database (clean)	1353
Database (dirty)	38
I/O	0
Latched	0

2008 Output

Buffer Pool	Value
Committed	39664
Target	384000
Database	14582
Dirty	1551
In IO	0
Latched	0
Free	18947
Stolen	6135
Reserved	0
Visible	384000
Stolen Potential	358665
Limiting Factor	17
Last OOM Factor	0
Page Life Expectancy	1267

2012 Output

Memory Manager	KB
VM Reserved	29551632
VM Committed	745624
Locked Pages Allocated	0
Large Pages Allocated	0
Emergency Memory	1024
Emergency Memory In Use	16
Target Committed	7786376
Current Committed	745624
Pages Allocated	203960
Pages Reserved	0
Pages Free	345136
Pages In Use	336504
Page Alloc Potential	8087616
NUMA Growth Phase	0
Last OOM Factor	6
Last OS Error	0

* Page allocations in 2012

DBCC MEMORYSTATUS

- A high percentage (>75-80%) of **stolen pages** relative to **target** is an indicator of the internal memory pressure.

2005 Output

Buffer Counts	Buffers
Committed	1064
Target	17551
Hashed	345
Stolen Potential	121857
External Reservation	645
Min Free	64
Visible	17551
Available Paging File	451997
Buffer Distribution	**Buffers**
Stolen	553
Free	103
Cached	161
Database (clean)	1353
Database (dirty)	38
I/O	0
Latched	0

2008 Output

Buffer Pool	Value
Committed	39664
Target	384000
Database	14582
Dirty	1551
In IO	0
Latched	0
Free	18947
Stolen	6135
Reserved	0
Visible	384000
Stolen Potential	358665
Limiting Factor	17
Last OOM Factor	0
Page Life Expectancy	1267

2012 Output

Memory Manager	KB
VM Reserved	29551632
VM Committed	745624
Locked Pages Allocated	0
Large Pages Allocated	0
Emergency Memory	1024
Emergency Memory In Use	16
Target Committed	7786376
Current Committed	745624
Pages Allocated	203960
Pages Reserved	0
Pages Free	345136
Pages In Use	336504
Page Alloc Potential	8087616
NUMA Growth Phase	0
Last OOM Factor	6
Last OS Error	0

DBCC MEMORYSTATUS

- **Stolen Potential** shows the maximum pages that can be stolen from the BP.
- Stolen + database pages are shown as **Page Alloc Potential** in SQL Server 2012.

2005 Output

Buffer Counts	Buffers
Committed	1064
Target	17551
Hashed	345
Stolen Potential	121857
External Reservation	645
Min Free	64
Visible	17551
Available Paging File	451997
Buffer Distribution	**Buffers**
Stolen	553
Free	103
Cached	161
Database (clean)	1353
Database (dirty)	38
I/O	0
Latched	0

2008 Output

Buffer Pool	Value
Committed	39664
Target	384000
Database	14582
Dirty	1551
In IO	0
Latched	0
Free	18947
Stolen	6135
Reserved	0
Visible	384000
Stolen Potential	358665
Limiting Factor	17
Last OOM Factor	0
Page Life Expectancy	1267

2012 Output

Memory Manager	KB
VM Reserved	29551632
VM Committed	745624
Locked Pages Allocated	0
Large Pages Allocated	0
Emergency Memory	1024
Emergency Memory In Use	16
Target Committed	7786376
Current Committed	745624
Pages Allocated	203960
Pages Reserved	0
Pages Free	345136
Pages In Use	336504
Page Alloc Potential	8087616
NUMA Growth Phase	0
Last OOM Factor	6
Last OS Error	0

DBCC MEMORYSTATUS

- **Free** shows committed buffers that are not currently being used.
- Other components may request these buffers and then mark these buffers as **Stolen**.

2005 Output

Buffer Counts	Buffers
Committed	1064
Target	17551
Hashed	345
Stolen Potential	121857
External Reservation	645
Min Free	64
Visible	17551
Available Paging File	451997
Buffer Distribution	**Buffers**
Stolen	553
Free	103
Cached	161
Database (clean)	1353
Database (dirty)	38
I/O	0
Latched	0

2008 Output

Buffer Pool	Value
Committed	39664
Target	384000
Database	14582
Dirty	1551
In IO	0
Latched	0
Free	18947
Stolen	6135
Reserved	0
Visible	384000
Stolen Potential	358665
Limiting Factor	17
Last OOM Factor	0
Page Life Expectancy	1267

2012 Output

Memory Manager	KB
VM Reserved	29551632
VM Committed	745624
Locked Pages Allocated	0
Large Pages Allocated	0
Emergency Memory	1024
Emergency Memory In Use	16
Target Committed	7786376
Current Committed	745624
Pages Allocated	203960
Pages Reserved	0
Pages Free	345136
Pages In Use	336504
Page Alloc Potential	8087616
NUMA Growth Phase	0
Last OOM Factor	6
Last OS Error	0

Logged memory errors

- **Failed Virtual Allocate Bytes: FAIL_VIRTUAL_RESERVE <size>**
- **Failed Virtual Allocate Bytes: FAIL_VIRTUAL_COMMIT <size>**
- 701 - There is insufficient system memory to run this query.
- 701 - There is insufficient system memory in resource pool 'default' to run this query.
- 802 - There is insufficient memory available in the buffer pool.
- 8628 - A time out occurred while waiting to optimize the query. Rerun the query.
- 8645 - A time out occurred while waiting for memory resources to execute the query. Rerun the query.
- 8651 - Could not perform the requested operation because the minimum query memory is not available. Decrease the configured value for the 'min memory per query' server configuration option.

Manually Releasing Memory

DBCC FREEPROCCACHE

- [(plan_handle | sql_handle | 'RG Pool Name')]

Free up all of proc cache, specific cache object, or RG pool

DBCC DROPCLEANBUFFERS

Put any non-dirty buffer on free list

DBCC FREESYSTEMCACHE

- ('ALL' | 'GarbageCollect' | <CacheName>, ['RG Pool Name'])

Free cache and/or user stores

DBCC FLUSHPROCINDB(<dbid>)

Free proc cache for DB (undoc.)

DBCC FREESESSIONCACHE

Free distributed query conn cache
(MEMOBJ_REMOTESESSIONCACHE)

Summary

- Foundations of memory related concepts
- Types of memory pressure
- How SQL Server engine deals with memory pressure internally
- What Tools, performance counters and DMVs to leverage, in order to troubleshoot memory related bottlenecks

Q&A

- http://blogs.msdn.com/b/blogdoezequiel

Obrigado!

Microsoft

Rumos
Formação Profissional