

SQL Server 2008

Performance Troubleshooting Part 2

Focus Area: Understanding Memory Internals



Chennai SQL Server User Group

Blog: <http://sql-articles.com/category/cssug/>

Ramkumar Gopal

Living For SQL Server

www.livingforsqlserver.com



<http://www.facebook.com/LivingForSqlServer>

Chennai SQL Server User Group (CSSUG) Meet – 23 Feb 2013

Agenda

- Windows Memory Concepts & Terminologies
- SysInternals Utilities
 - Process Explorer
 - VMMap
 - RAMMap
 - TestLimit.exe
- Physical Memory Internals
- Explore Physical Memory (RAMMap)
- Explorer Process Memory (VMMap)

Downloads

- Sysinternals Suite Download

<http://technet.microsoft.com/en-us/sysinternals/bb842062.aspx>

- TestLimit.exe (Sysinternals) Download

<http://download.sysinternals.com/files/TestLimit.zip>

- White Paper: SQL Server 2008 Troubleshooting Performance Problems

[http://msdn.microsoft.com/en-us/library/dd672789\(v=sql.100\).aspx](http://msdn.microsoft.com/en-us/library/dd672789(v=sql.100).aspx)

- Quest PerfMon Counters Poster

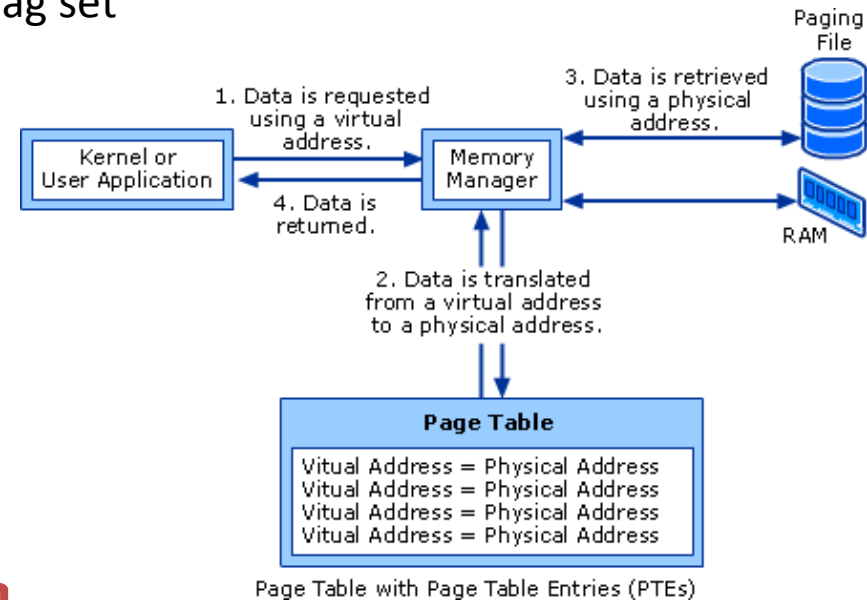
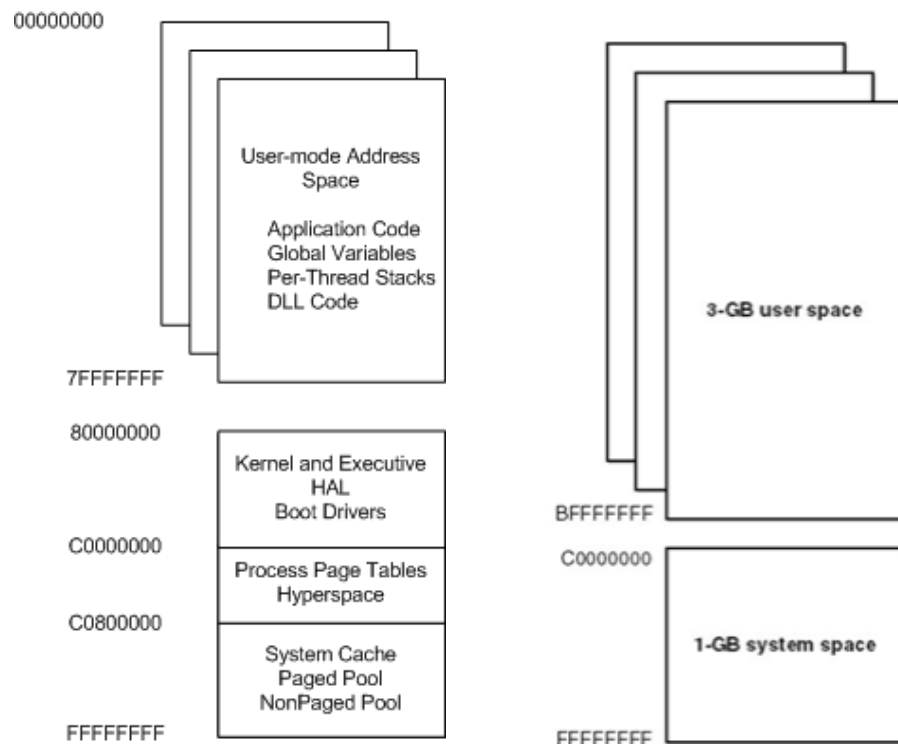
<http://www.quest.com/backstage/images/promotions/SQLServer-Perfmonance-Poster.pdf>

Recollecting Memory Concepts

- x86, x64 Virtual Memory Limits (3GB, PAE, AWE)
- Terminologies
 - Virtual Memory
 - Commit Limit, Peak & Current Limits
 - Process VAS (Private/Shared bytes)
 - Process Working Set (Private/Shared bytes)
 - Standby/Modified/Free/Zero page lists
 - Working Set Trimming
 - Soft/Hard Page Faults

32 bit Virtual Address Space

- $2^{32} = 4$ GB. Memory logically divided as 4 KB pages
- Each process gets 4 GB VAS
- 2 GB User Mode and 2 GB Kernel Mode
- User Mode VAS is configurable between 2 GB and 3 GB
/3GB or /USERVA options in boot.ini
IMAGE_FILE_LARGE_ADDRESS_AWARE flag set

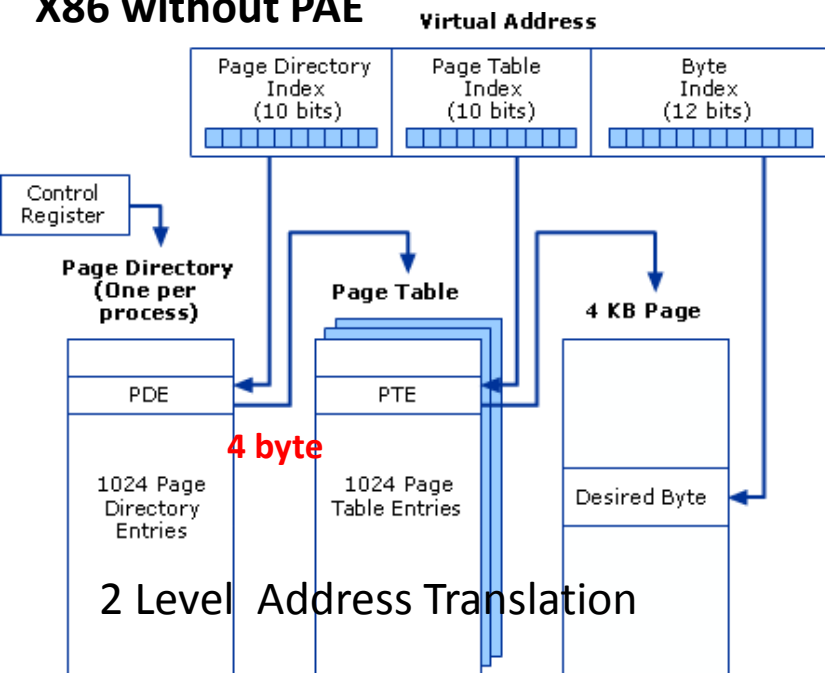


System PTE 251,980 (1 GB) to 34,884 (140 MB)
Non Paged Pool 200 MB to 130 MB

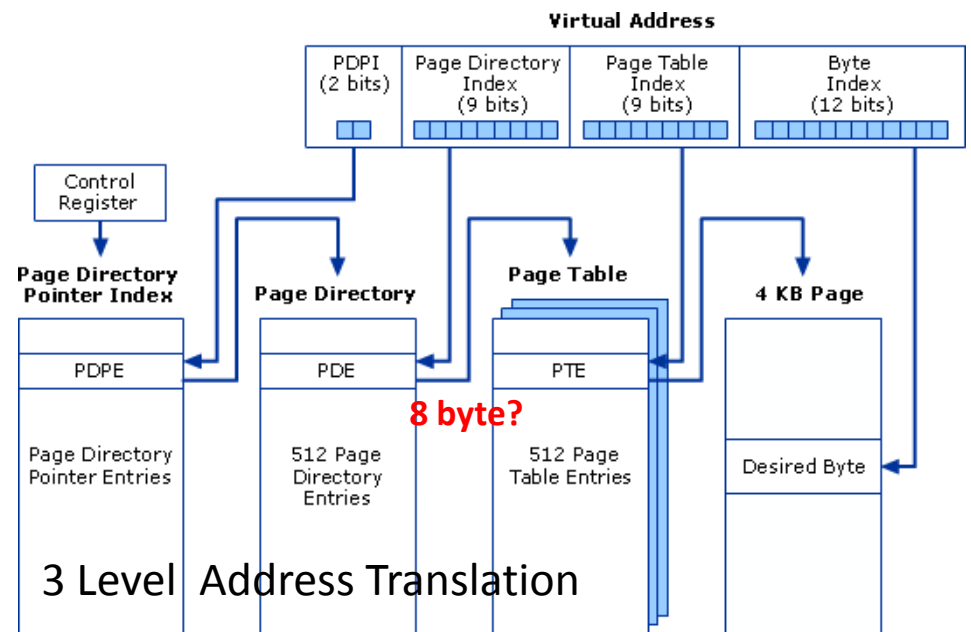
X86 - Physical Address Extensions (PAE)

- Only supported on 32 bit Windows
- Enable by adding /PAE switch in boot.ini
- Allow access to physical memory larger than 4 GB (PTE size from 4 KB to 8 KB)
- 4 GT tuning (/3gb switch) supported up to 16 GB
- 32 bit address of VAS remains unchanged
- Special mechanism for processes to access > 4 GB VAS (AWE APIs)

X86 without PAE



X86 with PAE $(4 * (512 * 4) * (512 * 4) * 4096) = 64 \text{ GB?}$



X86 - Application Windowing Extensions (AWE)

(AWE) is a set of extensions that allows an application to quickly manipulate physical memory greater than 4GB.

An application must have the Lock Pages in Memory privilege to use AWE.

```
---  
---  
if( ! LoggedSetLockPagesPrivilege( GetCurrentProcess(), TRUE )  
---  
---  
ADD=VirtualAlloc(lpaddr,size,MEM_RESERVE | MEM_PHYSICAL,PAGE_REA  
---  
---  
bResult=AllocateUserPhysicalPages(GetCurrentProcess(),&sizemap,  
---  
---  
bResult=MapUserPhysicalPages(ADD,sizemap,aRAMPages);  
---  
---
```

In 32 bit System:

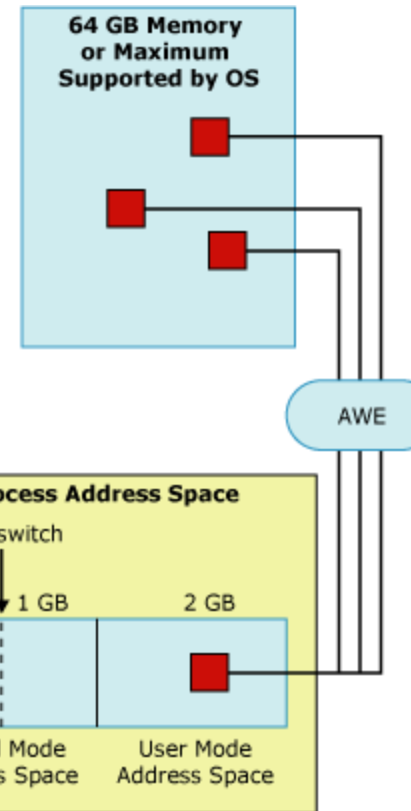
If RAM > 16 GB

Kernel Address Space Requires 2 GB

Don't enable /3GB in boot.ini

In 64 bit System

No effect on AWE. Lock pages in memory allowed



64 bit Virtual Address Space

Theory: $2^{64} = 17$ billion Terabytes.

Implementation: $2^{44} = 16$ TB (processor manufacturers)

8 TB User Mode Address Space

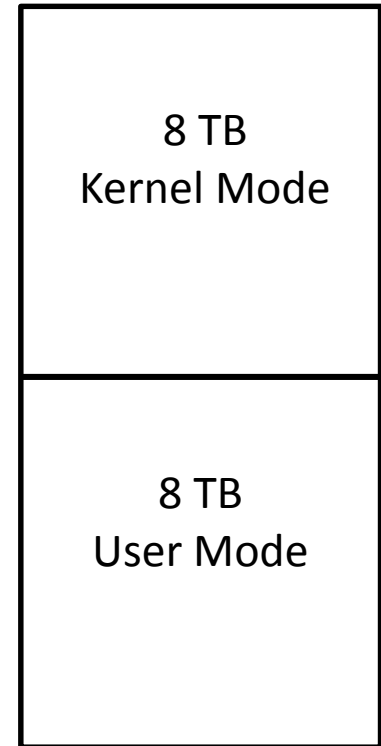
8 TB Kernel Mode Address Space

/3GB – Not Required

/PAE – Not Required/Supported

No effect on AWE. Lock pages in memory allowed

Virtual Address Space



A: Yes, we do recommend to turn on Lock pages in memory so that OS doesn't page SQL Server out. However on 64 bit you only need to grant the right "Lock Pages in Memory" to the SQL account for SQL Server to utilize this feature. You do need to change any of AWE settings through sp_configure.

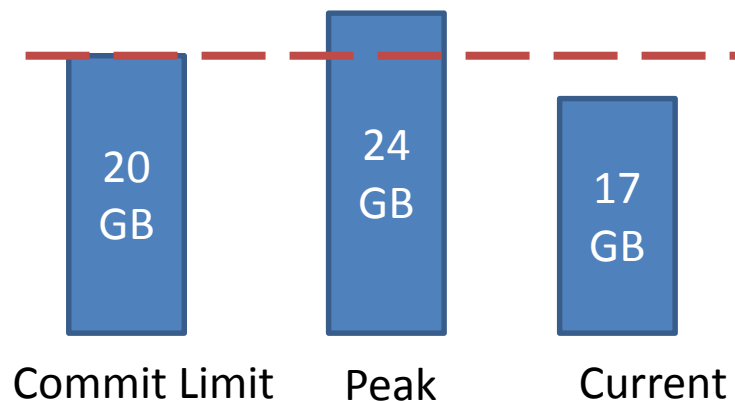
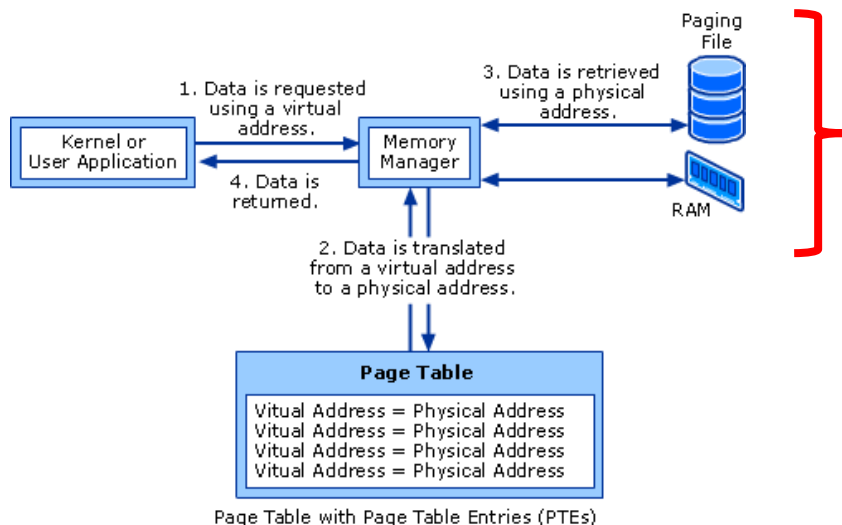
<http://blogs.msdn.com/b/slavao/archive/2005/11/15/493019.aspx>

Terminologies

- ✓ Windows Virtual Memory
- ✓ Commit Limit, Peak & Current Limits
- ✓ Process Virtual Memory & Private Bytes
- ✓ Working Set
- ✓ Standby/Dirty/Zero/Free Page lists
- ✓ Working Set Trimming
- ✓ Soft/Hard Page Faults

Virtual Memory

Virtual Memory = ~RAM + Max. Configured Size of Paging files



What does this mean?

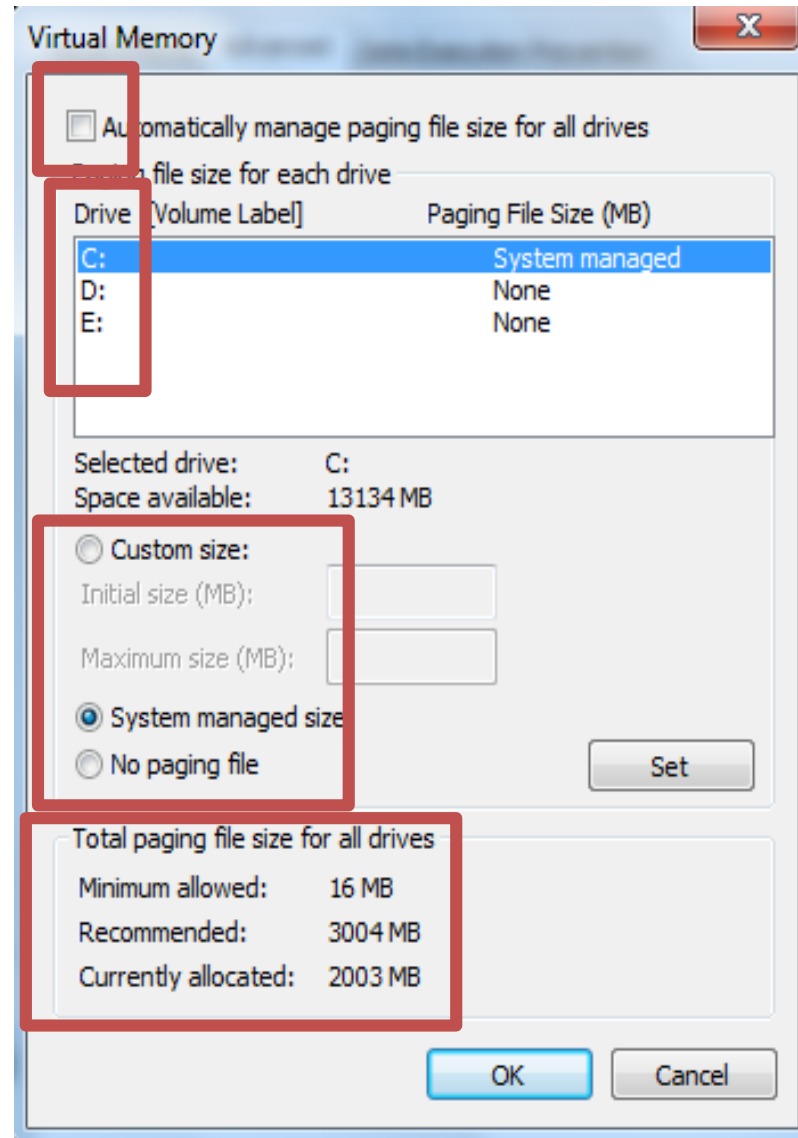
Commit Limit = Pageable RAM + Page File(s) Size

Current Limit = Current Commit charge in use

Peak = Highest Commit reached since server restart

Note: Peak a good memory health indicator

Page File Configurations



Process Explorer

System Information

Commit Charge (K)

Current	1,529,996
Limit	4,102,488
Peak	2,048,172

Peak/Limit	49.93%
Current/Limit	37.29%

Physical Memory (K)

Total	2,051,244
Available	931,640
Cache WS	76,820
Kernel WS	10,116
Driver WS	5,484

Kernel Memory (K)

Paged WS	169,108
Paged Virtual	235,012
Paged Limit	no symbols

Nonpaged	50,368
Nonpaged Limit	no symbols

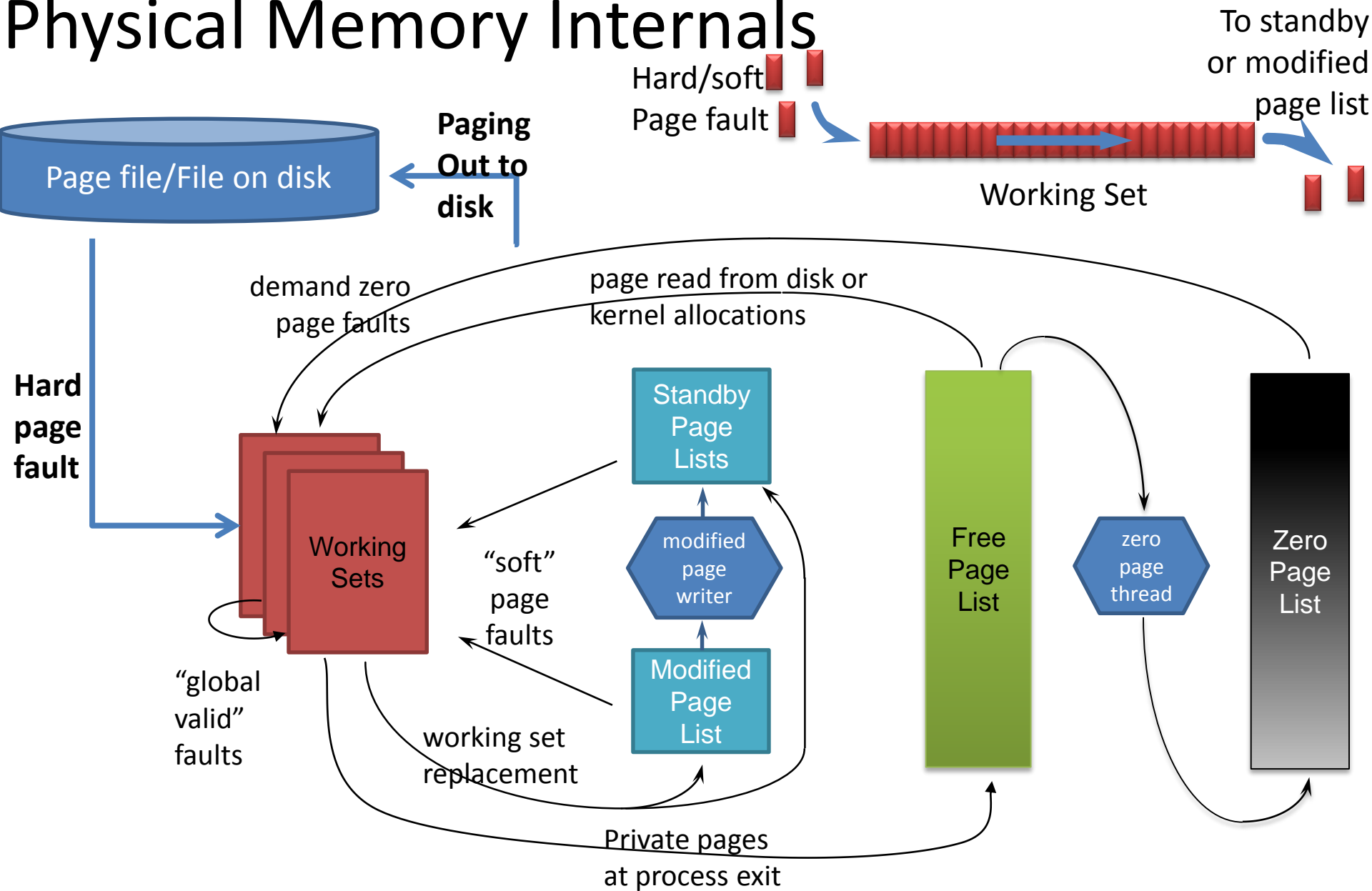
Paging

Page Fault Delta	0
Page Read Delta	0
Paging File Write Delta	0
Mapped File Write Delta	0

Paging Lists (K)

Zeroed	97,076
Free	16
Modified	30,792
ModifiedNoWrite	20
Standby	834,548
Priority 0	52
Priority 1	4,136
Priority 2	27,872
Priority 3	4,992
Priority 4	333,640
Priority 5	443,736
Priority 6	1,292
Priority 7	18,828

Physical Memory Internals



Reference & Courtesy : Mark Russinovich

Source: Mysteries of Windows Memory Management Revealed – Tech-Ed Europe 2010

Working set trimming

condition wherein Working Sets are flushed to the disk

Might be a BAU or an alarming issue in Physical Memory.

Page Faults

Occurs when requested pages are not in process Working set.

[Not Good]

Hard page fault – Read pages from page file/file on disk

[Good]

Soft page fault – Read pages from standby/modified page list

RAMMap (Physical Memory Consumers)

RamMap - Sysinternals: www.sysinternals.com

Usage	Total	Active	Standby	Modified	Zeroed	Free
Process Private	509,724 K	391,692 K	74,980 K	43,052 K		
Mapped File	773,352 K	193,336 K	578,744 K	1,272 K		
Shared Memory	90,212 K	67,376 K	5,912 K	16,924 K		
Page Table	19,452 K	19,064 K	188 K	200 K		
Paged Pool	151,448 K	94,572 K	21,352 K	35,524 K		
Nonpaged Pool	94,588 K	94,580 K				
System PTE	26,104 K	25,756 K	312 K	36 K		
Session Private	34,344 K	27,624 K	1,628 K	5,092 K		
Metafile	47,152 K	19,680 K	27,108 K			
AWE						
Driver Locked	42,896 K	42,896 K				
Kernel Stack	8,424 K	7,560 K	264 K	600 K		
Unused	253,548 K				246,396 K	7,152 K
Total	2,051,244 K	984,136 K	710,488 K	102,700 K	246,396 K	7,152 K

Available Bytes (Low-for long time indication of Memory pressure):

Stand by list + Free Page List + Zero Page List

Available Bytes

Commit Charge (K)		Kernel Memory (K)		Paging Lists (K)	
Current	1,529,996	Paged WS	169,108	Zeroed	97,076
Limit	4,102,488	Paged Virtual	235,012	Free	16
Peak	2,048,172	Paged Limit	no symbols	Modified	30,792
Peak/Limit	49.93%	Nonpaged	50,368	ModifiedNoWrite	20
Current/Limit	37.29%	Nonpaged Limit	no symbols	Standby	834,548
Physical Memory (K)		Paging		Priority 0	52
Total	2,051,244	Page Fault Delta	0	Priority 1	4,136
Available	931,640	Page Read Delta	0	Priority 2	27,872
Cache WS	76,820	Paging File Write Delta	0	Priority 3	4,992
Kernel WS	10,116	Mapped File Write Delta	0	Priority 4	333,640
Driver WS	5,484			Priority 5	443,736
				Priority 6	1,292
				Priority 7	18,828

What do these values convey?

Process Explorer

Process Memory Consumption (high level)

Process Explorer - Sysinternals: www.sysinternals.com [HCLISD\ramkumar-g]

File Options View Process Find Handle Users Help

Process	Virtual Size	Private Bytes	Working Set	WS Private	WS Shareable	WS Shared
sqlservr.exe	1,808,412 K	105,116 K	37,232 K	33,096 K	4,136 K	1,792 K
ieexplore.exe	289,380 K	79,036 K	81,464 K	52,568 K	28,896 K	17,140 K
OUTLOOK.EXE	448,324 K	78,084 K	82,312 K	33,888 K	48,424 K	18,832 K
communicator.exe	277,184 K	73,496 K	33,876 K	20,888 K	12,988 K	6,776 K
Rtvscan.exe	636,856 K	73,360 K	6,104 K	2,844 K	3,260 K	2,544 K
dwm.exe	206,004 K	60,156 K	30,960 K	10,264 K	20,696 K	2,872 K
explorer.exe	315,284 K	59,540 K	59,956 K	34,140 K	25,816 K	18,648 K

Process Memory

Virtual Size – Total Non Free Pages Reserved by Process VAS
Private bytes (Data) + Shared bytes (Image files)

Private Bytes – Total size of Private Bytes (Mostly used by data)

Sharable Bytes – Total Size of memory that can be shared
with other processes (.Exe, .dll etc)

Working Set (WS) - Amount of Physical memory currently
used by process

Working Set Private –Physical Memory used by Process that
can not be shared

Working Set Shared –Physical Memory used by Process that
can be shared with other processes (exe, dll etc)

VMMMap

Process Memory Consumption (in detail)

VMMMap - Sysinternals: www.sysinternals.com

File Edit View Options Help

Process: sqlservr.exe
PID: 2596

Type	Size	Committed	Private	Total WS	Private WS	Shareable WS	Shared WS
Total	1,814,140 K	184,724 K	106,404 K	41,488 K	39,676 K	1,812 K	536 K
Image	78,384 K	78,384 K	7,276 K	2,220 K	476 K	1,744 K	524 K
Mapped File	5,528 K	5,528 K					
Shareable	3,892 K	1,620 K		60 K		60 K	8 K
Heap	8,512 K	3,044 K	2,980 K	412 K	408 K	4 K	
Managed Heap							
Stack	24,064 K	22,132 K	22,132 K	264 K	264 K		
Private Data	1,688,800 K	69,056 K	69,056 K	33,572 K	33,568 K	4 K	4 K
Page Table	4,960 K	4,960 K	4,960 K	4,960 K	4,960 K		
Free	287,908 K						

Testlimit

Warning: Don't try this command in Work environments.

C:\Windows\system32\cmd.exe

```
usage: testlimit [[-h [-u]] ; [-p [-n]] ; [-t [-n [KB]]] ; [-u [-i]] ;
ct size]] ; [-a|-d|-l|-m|-r|-s|-v [MB]] ; [-w]] [-c [count]] [-e [second
-a      Leak Address Windowing Extensions (AWE) memory in
        specified MBs (default is 1).
-c      Count of number of objects to allocate (default is as many a
        possible). This must be the last option specified.
-d      Leak and touch memory in specified MBs (default is 1).
-e      Seconds elapsed between allocations (default is 0).
-g      Create GDI handles of specified size (default 1 byte).
        Specify a size of 0 to cause GDI object exhaustion.
-h      Create handles. Specify -u to also allocate file objects.
-i      Exhaust USER desktop heap.
-l      Allocate the specified amount of large pages (rounded to lar
        size multiple.
-m      Leak memory in specified MBs (default is 1).
-p      Create processes - add -n to set min working set. Add -n to
        set min working set of processes to smallest.
-r      Reserve memory in specified MBs (default is 1).
-s      Leak shared memory in specified MBs (default is 1).
-t      Create threads - add -n to specify minimum stack reserve (in
-u      Create USER handles to menus.
-v      VirtualLock memory in specified MBs (default is 1).
-w      Reset working set minimum to highest possible value.
```

D:\>testlimit -m 1_

What Next?

- ✓ SQL Server Memory Consumers
- ✓ Troubleshooting External/Internal Memory Pressures

Questions