SQLOS – Basics

*SQL Server 2005, 2008, 2008 R2

Ramkumar Gopal

Living For SQL Server

Chennai SQL Server User Group

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

FB : www.facebook.com/groups/cssug

Google : chnsqlug@googlegroups.com

Twitter : www.twitter.com/sqlarticles

Blog: http://www.sqlservercentral.com/blogs/livingforsqlserver/

http://www.facebook.com/LivingForSqlServer

Chennai SQL Server User Group (CSSUG) Meet – 15 Dec 2012

Agenda

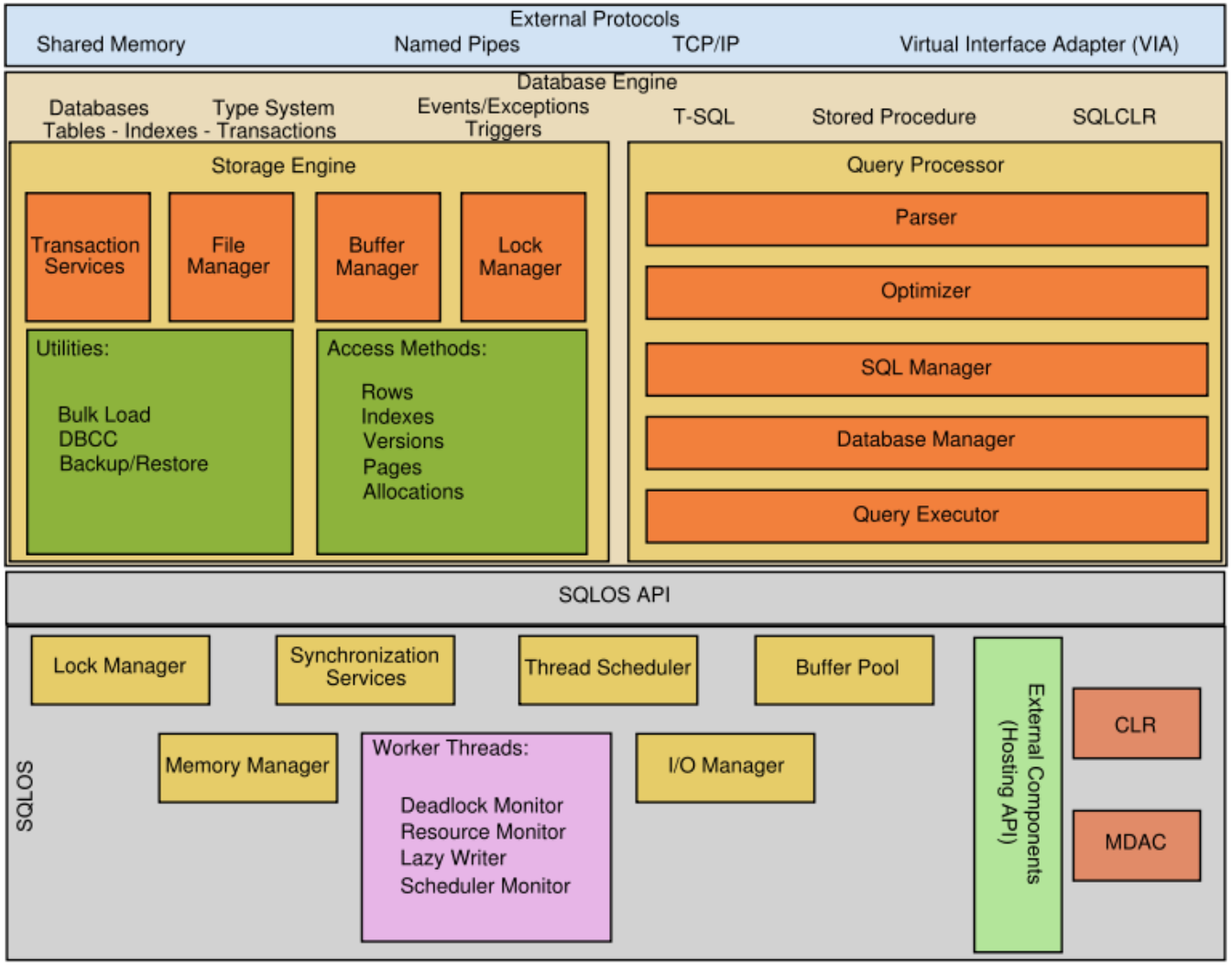
- SQL Server Architecture
- SQLOS - Basics
- SQLOS – Responsibilities
- CPU
  - SMP vs. NUMA
  - Preemptive vs. Non Preemptive Scheduling
- SQLOS & CPU
  - Requests – Tasks – Workers – Threads - Schedulers
- Memory
  - 32 bit vs. 64 bit Architecture, /3GB, PAE, AWE, LPM
- SQLOS & Memory

Reference & Courtesy:

- Bob Ward - PASS 2012 - Inside SQLOS 2012 Presentation
- Wrox Book: SQL Server 2008 Internals and Troubleshooting
- MS Press Book : SQL Server 2008 Internals
- mssqlwiki.com
- Slava Oks
- http://www.intellectualheaven.com/Articles/WinMM.pdf

Note: Source is mentioned in almost all the slides for further read

SQL Server Architecture



Source: http://blog.sqlauthority.com/2007/12/08/sql-server-generic-architecture-image/

SQLOS – Basics

- SQL Server 2005+
- User mode layer between SQL Server and the Windows OS.
- It is an engine in the form of two DLLs (SQLDK.DLL and SQLOS.DLL)
- Takes care of Scheduling and Memory Management functions
- Thread mode or fiber mode. Default is thread mode.

Source: Bob Ward - Inside SQLOS 2012 - PASS 2012

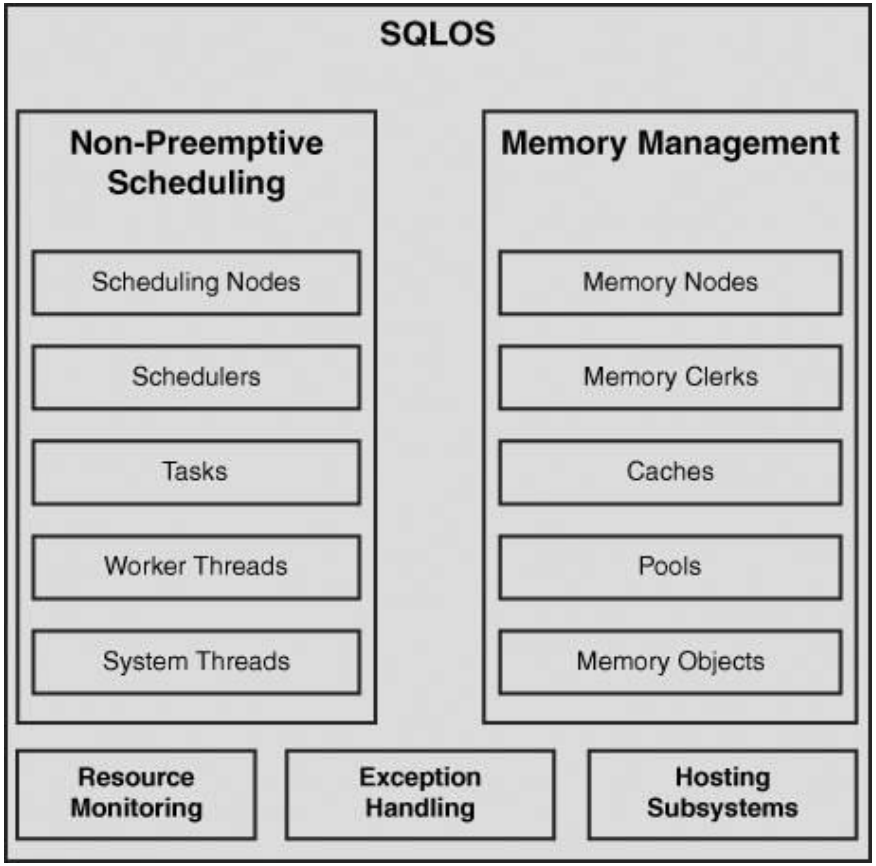
SQLOS – Basics (Cont..)

- Supports both SMP and NUMA
- Set of APIs (interface to the Operating System for SQL Server)
- Its RDBMS Independent
- SQL Server Uses Non-Preemptive Scheduling
- Provides mechanism for affinity to NUMA and CPUs
  No NUMA = NUMA node 0

Source: Bob Ward - Inside SQLOS 2012 - PASS 2012

SQLOS – Responsibilities

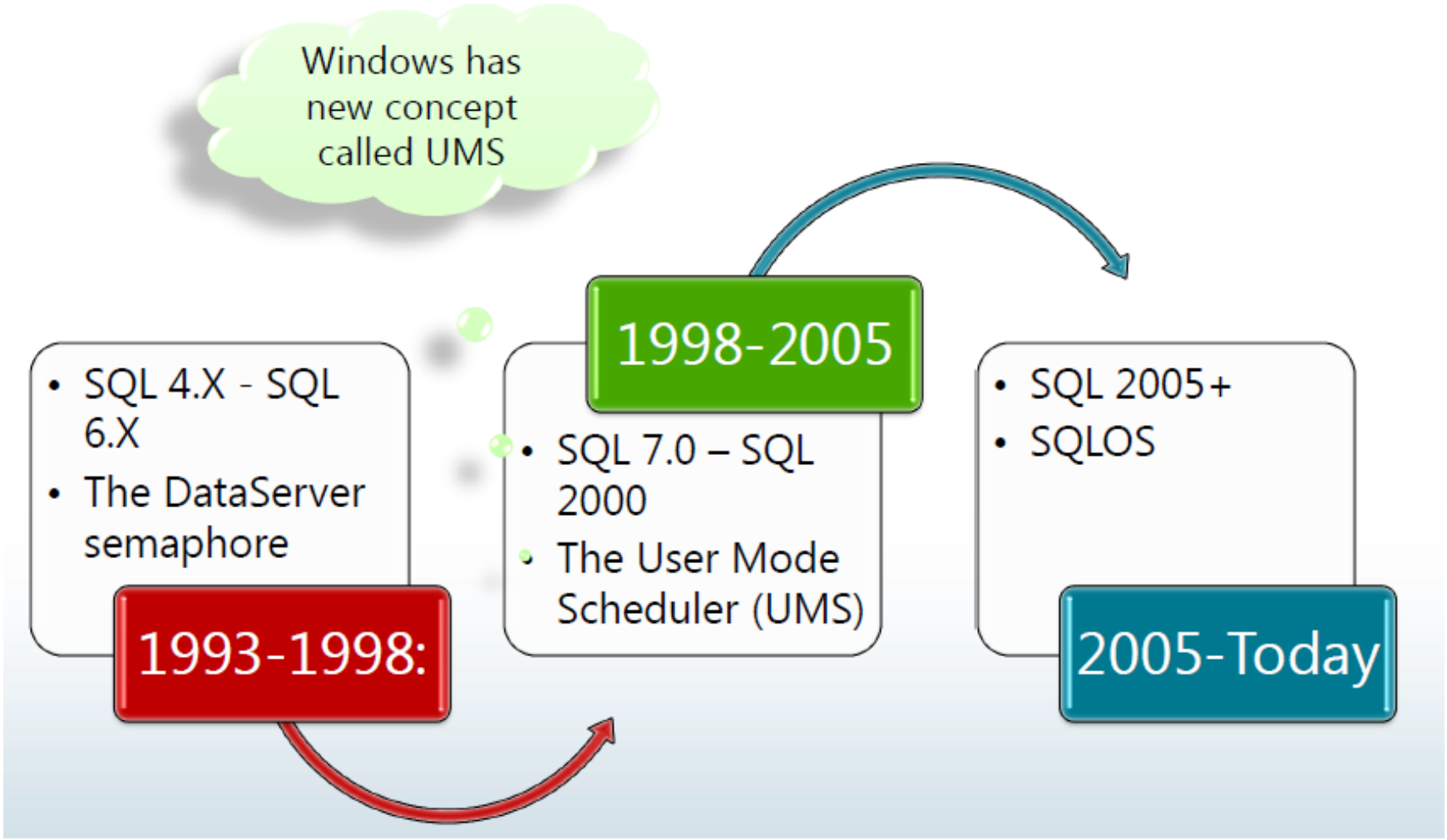
- **CPU Scheduling**
- **Memory Management**
- Extended Events
- Deadlock Detection
- SQL OS Exception Handling
- Synchronization



http://flylib.com/books/en/3.3.1.12/1/

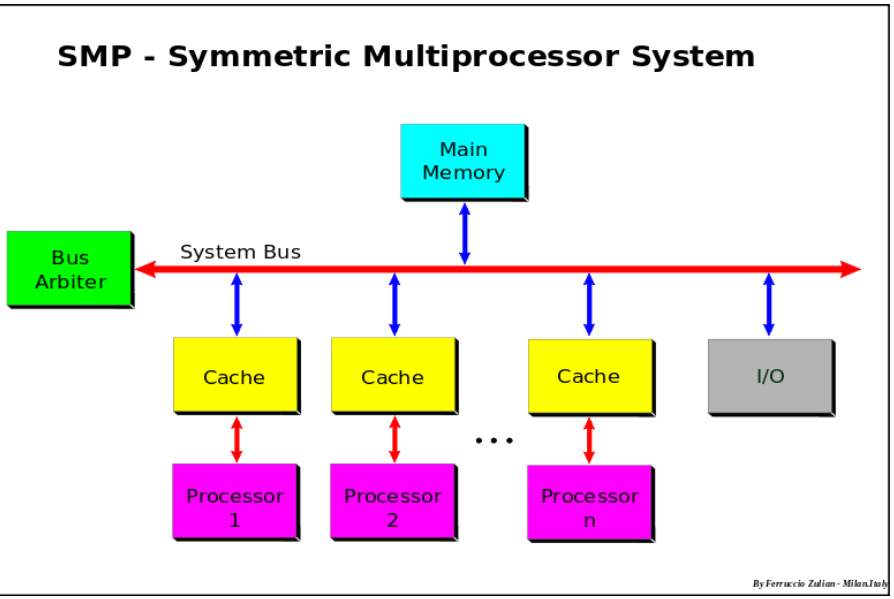
Source: MS Press SQL Server 2008 Internals (CH 1, Page 18)

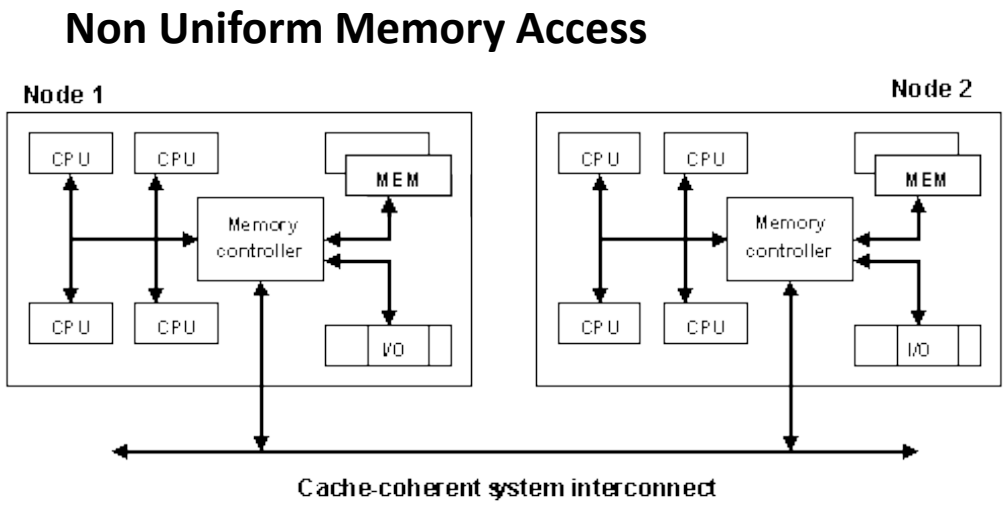
SQLOS – Journey So far



Source: Bob Ward - Inside SQLOS 2012 - PASS 2012

CPU – SMP vs NUMA





SMP - Multiple processors are connected on the system memory symmetrically and access the system memory equally and uniformly

When processors increases competition to access common/single memory bus increases

NUMA - logically divides into multiple nodes, which can access both local and remote memory resources

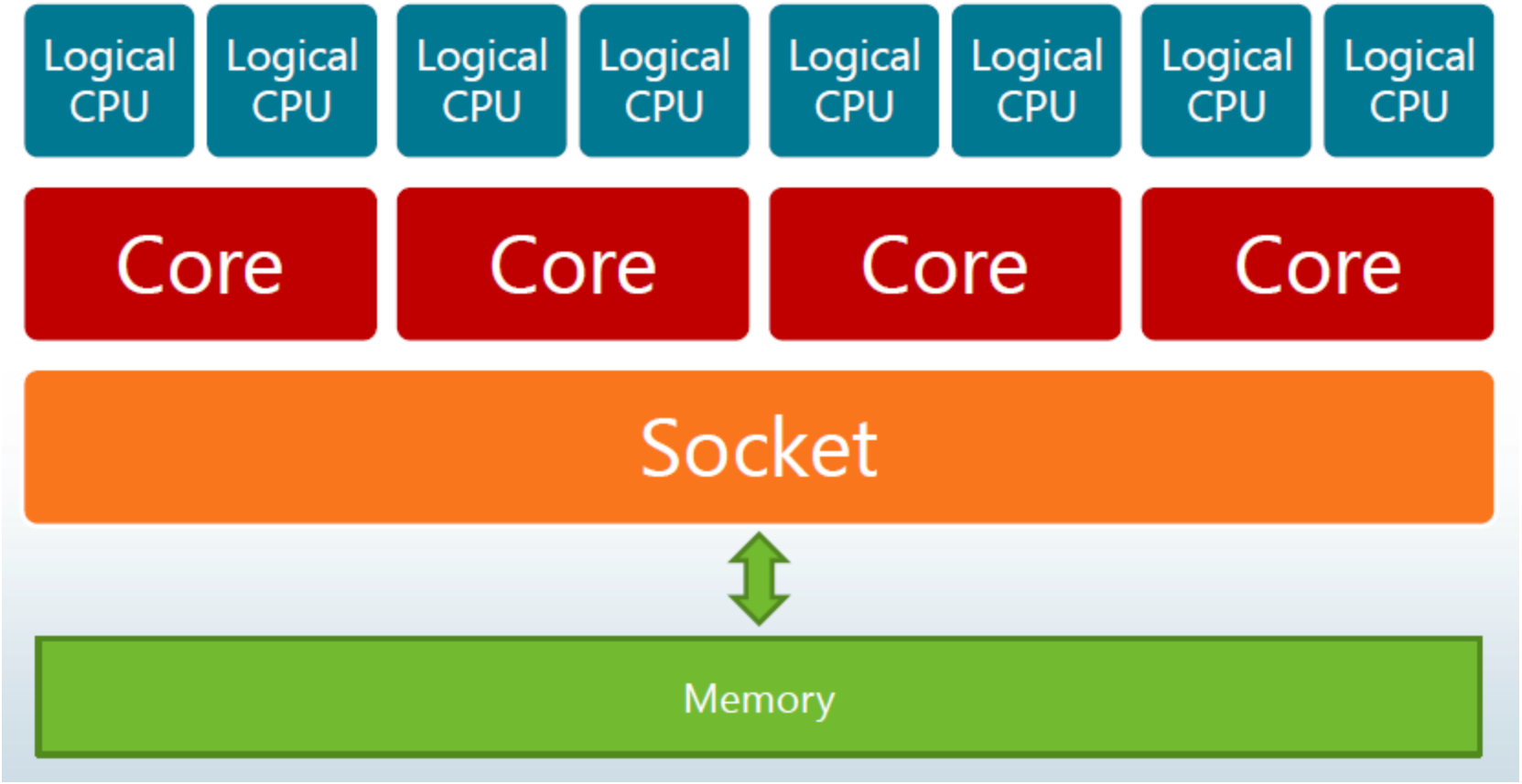
Accessing memory of another node is costly.

http://msdn.microsoft.com/en-us/library/windows/hardware/gg463257.aspx
http://en.wikipedia.org/wiki/Symmetric_multiprocessing
http://software.intel.com/en-us/blogs/2009/03/11/learning-experience-of-numa-and-intels-next-generation-xeon-processo

CPU – SMP

http://en.wikipedia.org/wiki/Hyper-threading

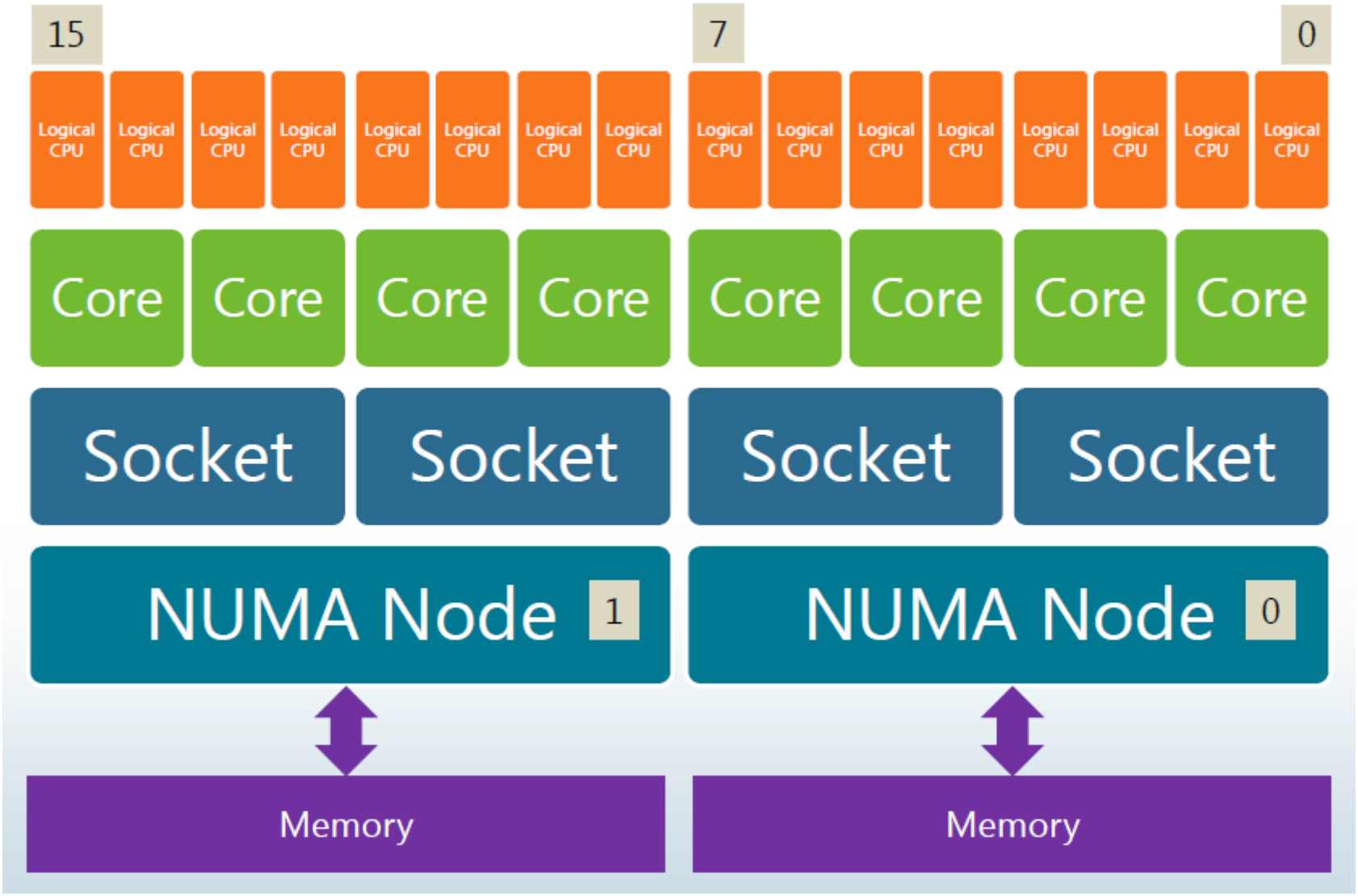
Hyper threading enabled
Intel – BIOS settings



Source: Bob Ward - Inside SQLOS 2012 - PASS 2012

CPU – NUMA

(Processor Groups, Nodes)



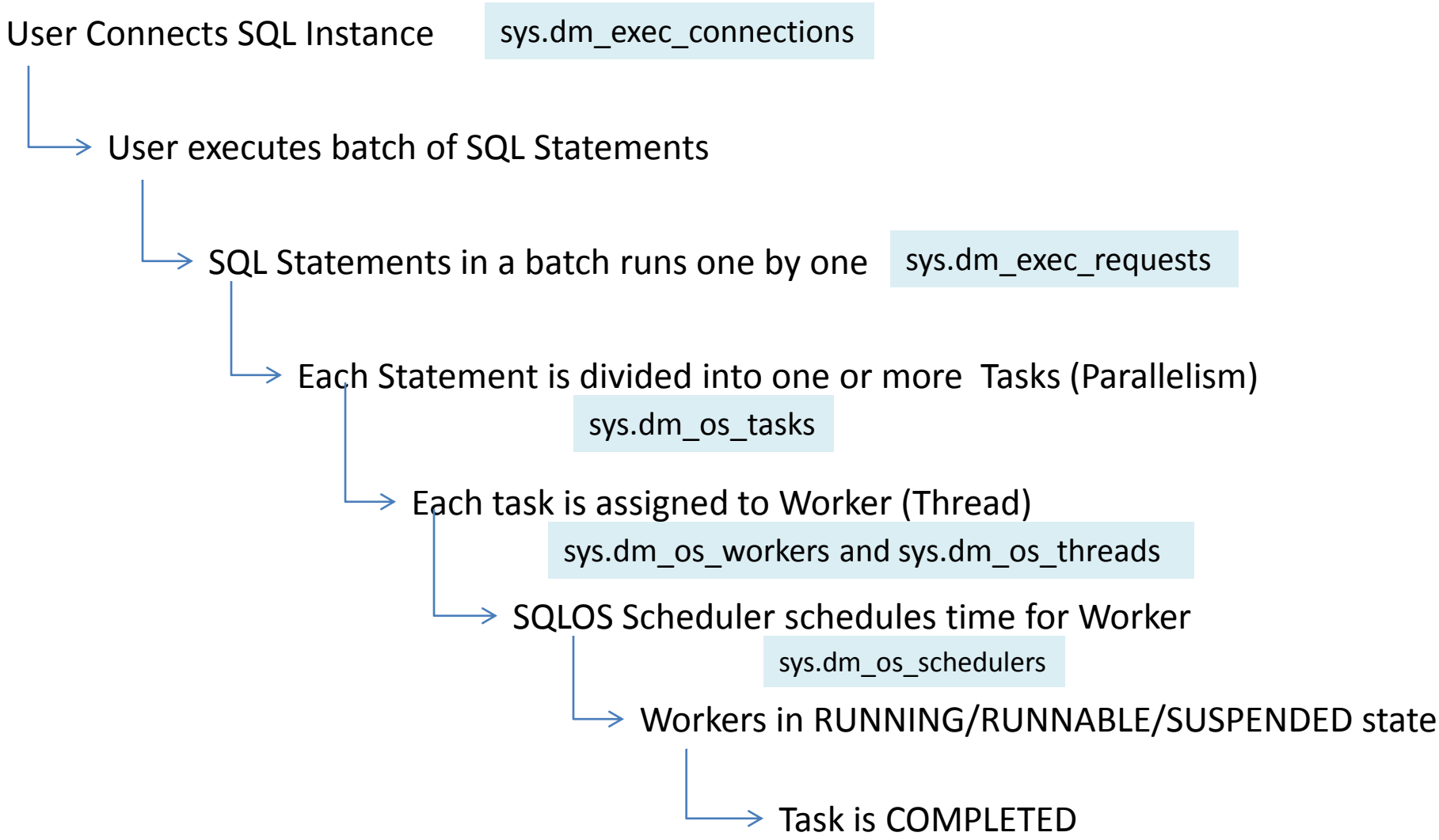
Source: Bob Ward - Inside SQLOS 2012 - PASS 2012

CPU - Preemptive vs Non Preemptive Scheduling

- Preemptive scheduling
  - A thread **can be interrupted** or preempted by another thread on the same CPU even while it is running.
  - This is the default behavior for threads running in a process for Windows 95 (32bit)/Windows NT and later.
- Non-preemptive scheduling
  - This is also known as *cooperative multitasking*
  - A thread runs until **it decides** to allow another thread to run
  - This was the behavior of Windows prior to Win95 such as Windows 3.1 (and in 16bit apps in Win95)

Source: Bob Ward - Inside SQLOS 2012 - PASS 2012

SQLOS – CPU and Query



http://www.sqlservercentral.com/blogs/livingforsqlserver/2012/12/03/sqlos-basics-query-and-cp

SQLOS – CPU and Query

DEMO

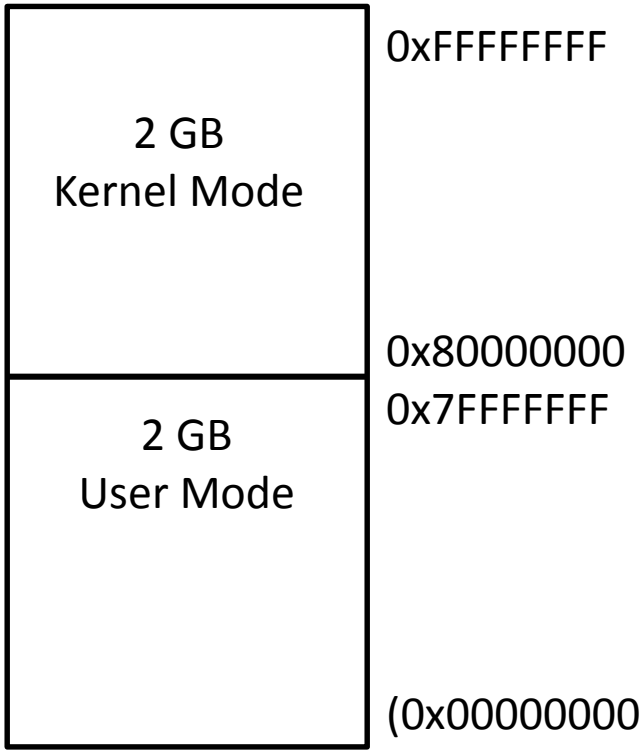
Memory - Basics

- If all the processes have access to physical memory, there would be a bottleneck in the system very quickly
- Windows assigns a virtual address space (VAS) to each process
- VAS is the abstraction layer between process and physical memory
- Size of VAS space is determined by CPU architecture (32 bit or 64 bit)
- 32 bit (2^32) = 4 GB
  - 2 GB Kernel mode, 2 GB User mode
- 64 bit (2^64) = 16 exabytes.
- In 64 bit architecture only 44 bits used 2^44 = 16 TB
  - 8 TB Kernel mode, 8 TB User mode

Source: Wrox – SQL Server 2008 Internals and Troubleshooting book

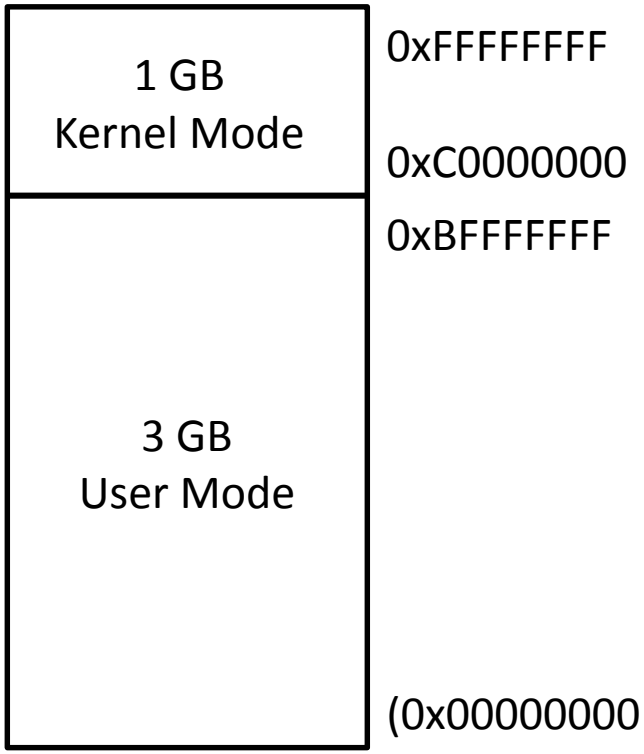
Memory - Virtual Memory Management in 32 bit

Virtual Address Space



/USERVA

Virtual Address Space **/3GB Switch**



/PAE – Access to memory between 4GB and up to 128*GB – 36 bits to address memory
AWE – How an application can reach beyond the 4GB limit, when using /PAE
AWE Allocated memory is nonpageable & Locked

Source: Wrox – SQL Server 2008 Internals and Troubleshooting book
http://blogs.technet.com/b/josebda/archive/2009/03/30/sql-server-2008-sqlos.aspx

Memory - Virtual Memory Manager

- Maps VAS with Physical Memory when process requires Read/Write.

- Process of temporarily moving unused data to disk is called **Paging (or Trimming?)**

- Move contents of a process from Page file (**Page Fault**)

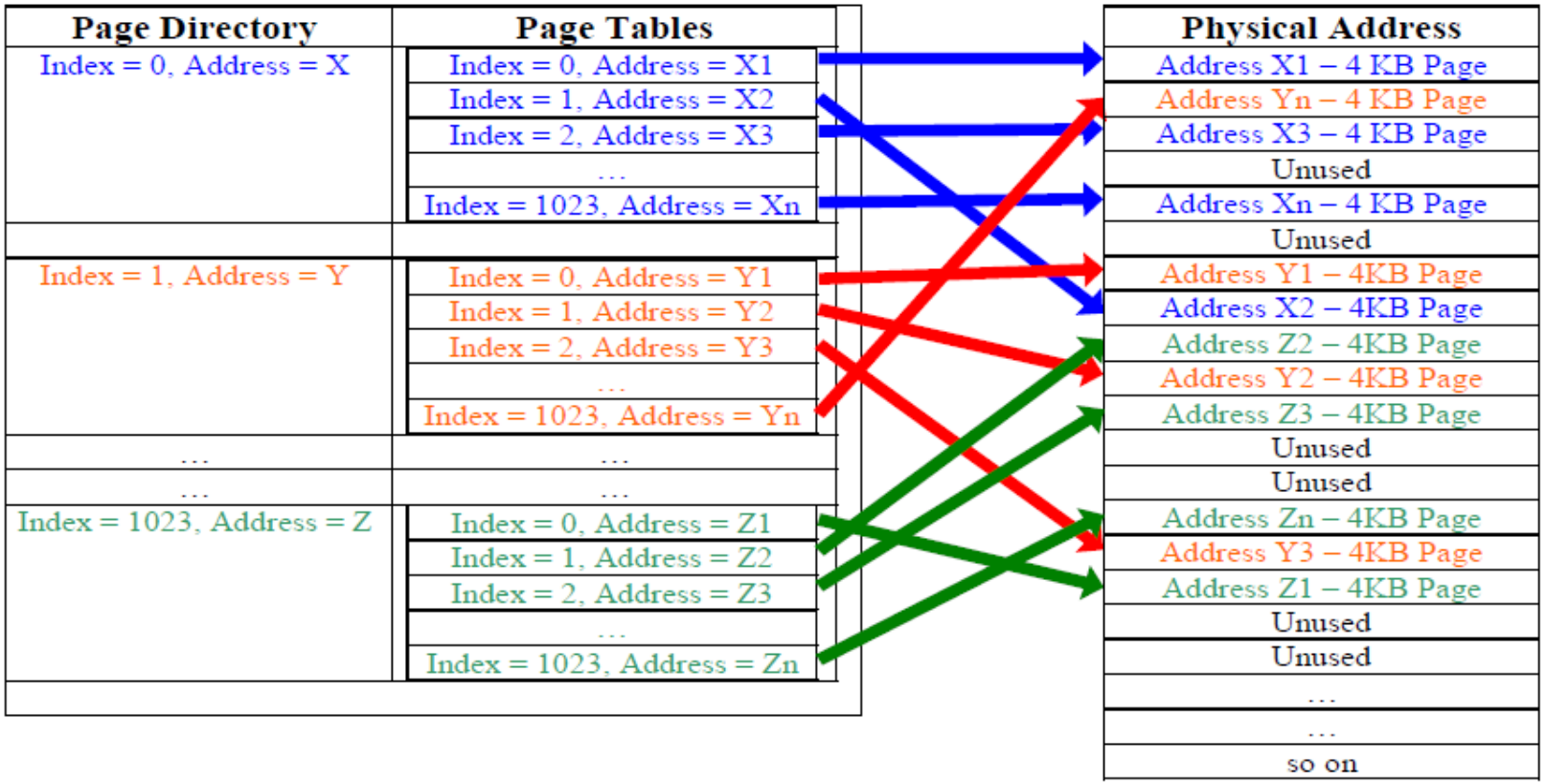
- **Working Set** - Portion of a process's VAS that currently maps to physical memory

- Requested data is not in Working set, data has to be loaded from Page file. This is called **Hard Page Fault**.
  Soft Page Fault – Page still on standby list of Physical Memory

Source: Wrox – SQL Server 2008 Internals and Troubleshooting book

Memory - Virtual Memory Manager (Cont..)

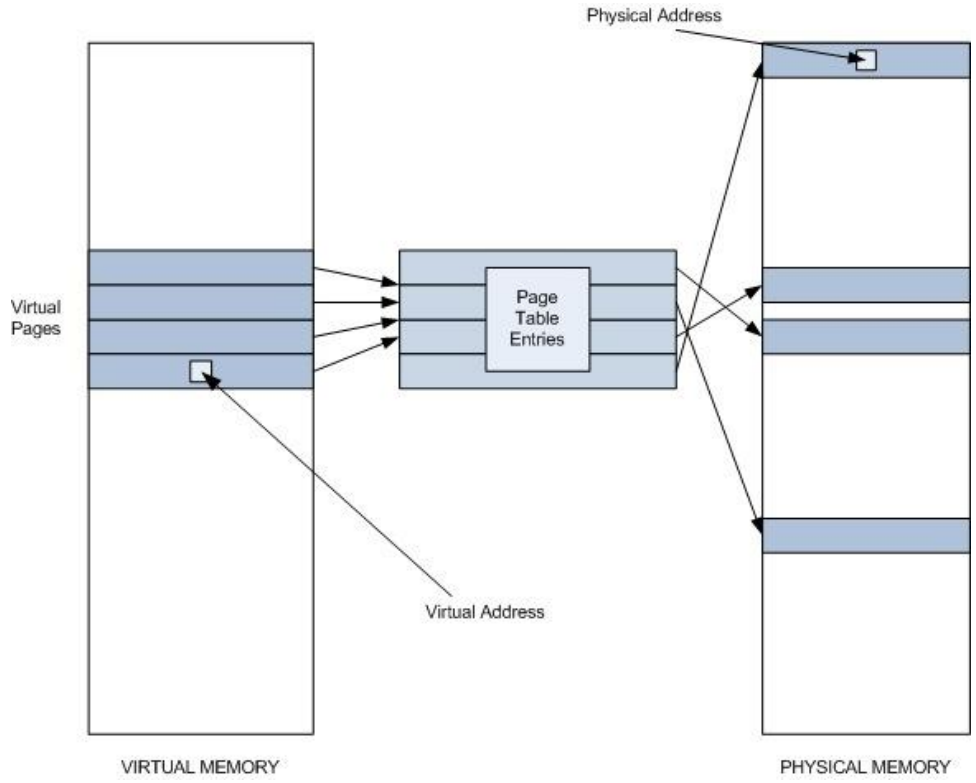
- The 32 bit processor divides the physical address space in 4 KB pages (Meaning 1024 X 1024 addresses to point 4 GB of 4 KB pages).
- Dimension 1 => PDE 1024 entries.
  Dimension 2 => PTE 1024 PTEs per PDE. (1024 * 1024 * 4 bytes) = **4 MB**



Source: http://www.intellectualheaven.com/Articles/WinMM.pdf

Memory – Virtual Memory to Physical Memory Mapping

PTE – Page Table Entry



Each process has its own Page Directory and Page Tables. Thus windows allocate 4 MB of this space per process

A 32 bit logical address

First 10 bits – PDE
Next 10 bits – PTE
Last 12 bits – To address page

Virtual Memory Manager

Tracks VAS Address and Physical Address Mapping using PTE

http://www.windowsitpro.com/content1/topic/inside-memory-management-part-1/catpath/internals-and-architecture
http://www.intellectualheaven.com/Articles/WinMM.pdf

Memory – Paged and Non Paged Pools

Paged Pools – System Memory region that can be paged out to disk

Non Paged Pools – System Memory regions that can not be paged out to disk.
Important hardware drivers may lead to system crash if paged to disk

Note:
/3GB

- Impacts Paged and Non Paged Pool size
- PTE count (Example, From 2,00,000 to 25,000 appx.)

/PAE modifies size of PTE from 4 KB to 8 KB
(Doubles PTE space requirement in Kernel mode address space)

It is not recommended to use both /PAE and /3GB

Source: Wrox – SQL Server 2008 Internals and Troubleshooting book

SQL Server - Memory

Memory Regions

Bpool and MemToLeave (and BUF)

Memory Manager Components

- Memory Nodes
- Memory Clerks
- Caches
- Pools
- Memory Objects

SQLOS Memory related DMVs

```
select * from sys.dm_os_memory_objects
select * from sys.dm_os_memory_pools
select * from sys.dm_os_memory_nodes
select * from sys.dm_os_memory_cache_entries
select * from sys.dm_os_memory_cache_hash_tables
```

SQL Server – Memory Regions

- SQL Server "User address space" is broken into two regions: **MemToLeave and Buffer Pool**
- Size of MemToLeave (MTL) and Buffer Pool (BPool) is determined by SQL Server during start up.

1. MTL (Memory to Leave) =

(Stack size * max worker threads) + Additional space
(By default 256 MB and can be controlled by -g).
Stack size =512 KB per thread for 32 Bit SQL Server
Ie = (256 *512 KB) + 256MB =384MB

MTL Consumers:

- ✓ **Memory allocations > 8 KB and need's contiguous memory (Multiple_pages_kb).**
- ✓ **COM Objects**
- ✓ **Extended Stored Procedures**
- ✓ **Dlls loaded in SQL Server Process**
- ✓ **Memory allocated by linked servers**
- ✓ **SQLCLR**

Source: http://mssqlwiki.com/sqlwiki/sql-performance/basics-of-sql-server-memory-architecture/

SQL Server – Memory Regions

2. Bpool

✓ Mostly Data and Index Pages - If Memory request is <= 8 KB
✓ Memory allocated by SQL Server memory manager for some memory clerk's

type
CACHESTORE_SQLCP
CACHESTORE_PHDR
MEMORYCLERK_SOSNODE
USERSTORE_DBMETADATA
USERSTORE_SCHEMAMGR
MEMORYCLERK_SQLGENERAL
MEMORYCLERK_SQLSTORENG
OBJECTSTORE_LBSS
MEMORYCLERK_SQLCLR
OBJECTSTORE_LOCK_MANAGER

3. BUF structures

SQL Server maintains a BUF structure for each page.
Used to track status information associated with each buffer

- ✓ Buffer latch
- ✓ Pointer to the actual 8 KB page
- ✓ Status bits that indicate whether the page is dirty
- ✓ Has an IO in progress etc.

Source: http://mssqlwiki.com/sqlwiki/sql-performance/basics-of-sql-server-memory-architecture/

SQLOS – Memory Manager Components

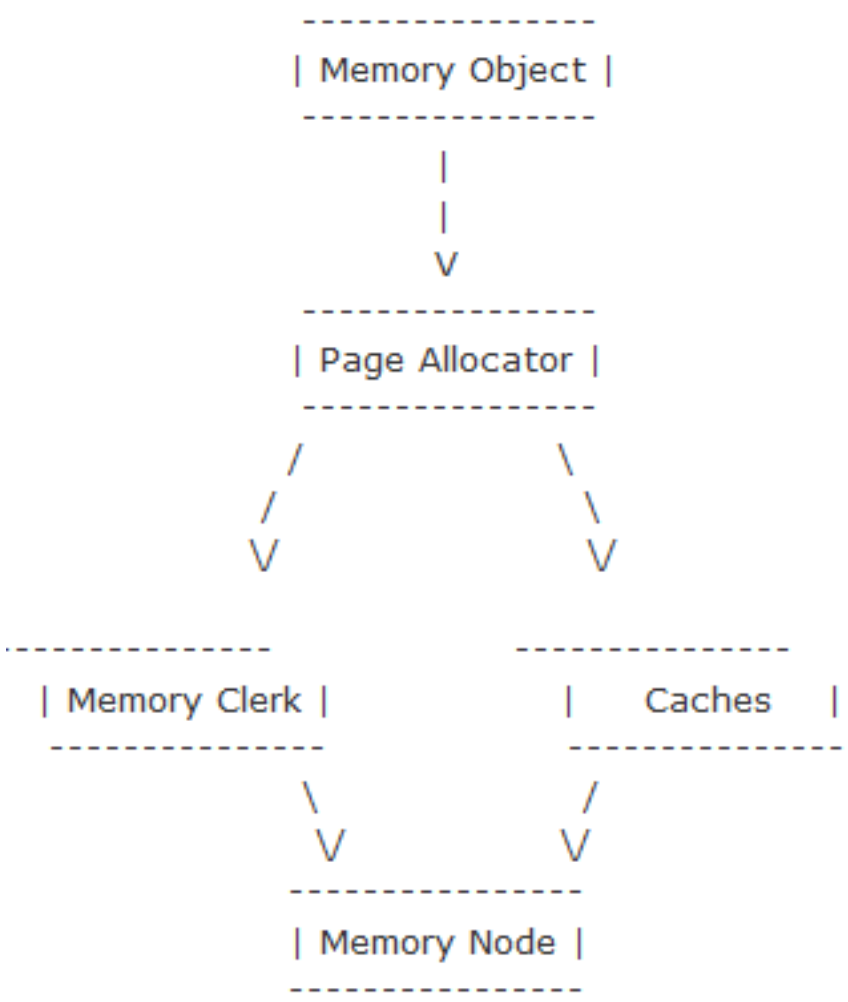
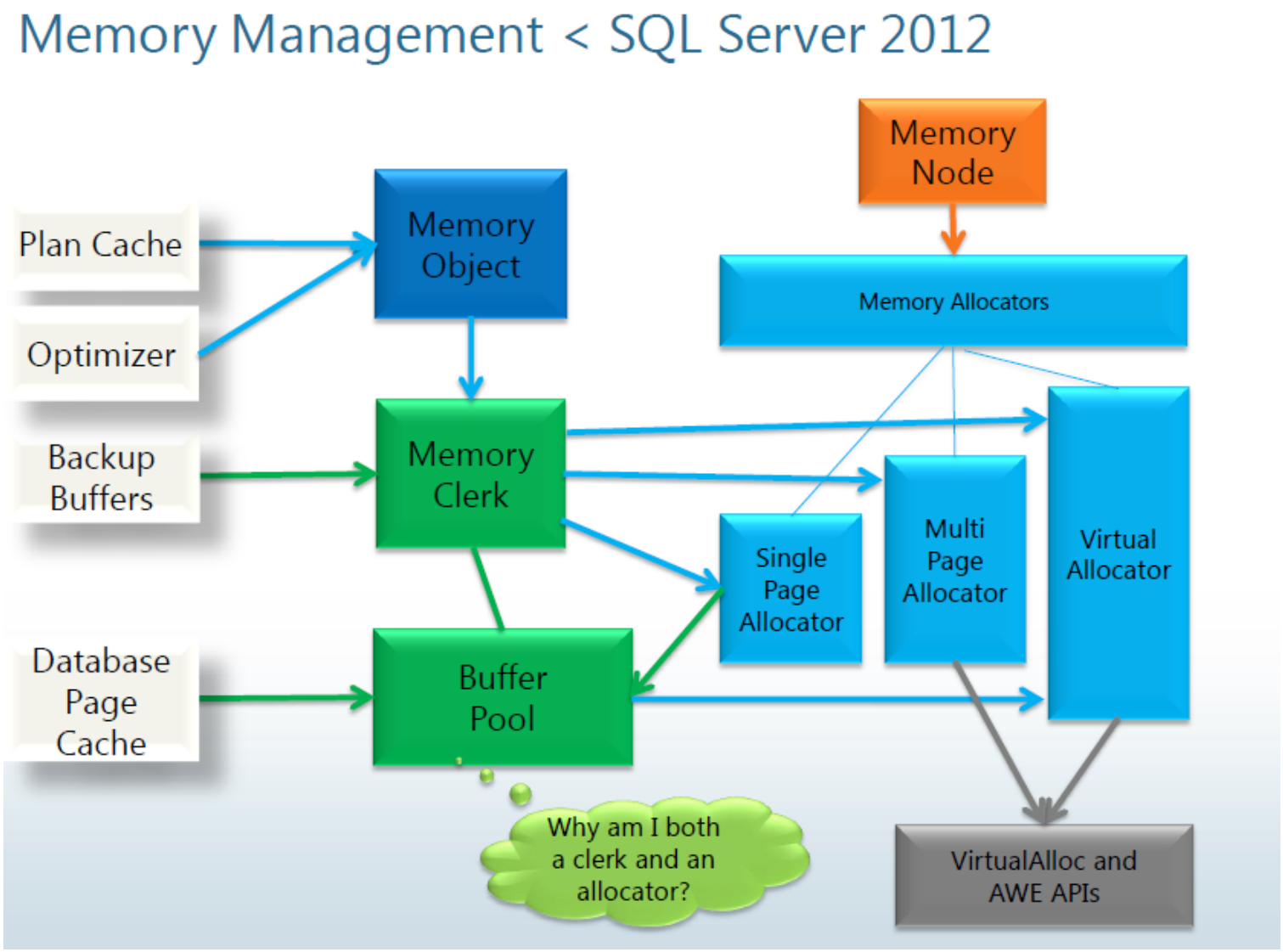


Fig. 1

http://blogs.msdn.com/b/slavao/archive/2005/02/11/371063.aspx

SQLOS – Memory Manager Components



Source: Bob Ward - Inside SQLOS 2012 - PASS 2012

SQLOS – Memory Manager Components

1. Memory Node (SQL Server Memory Allocators)

It is internal SQLOS's object

It consists of 3 major memory allocators

1. Page Allocators (Allocate memory in 8K size)
    - Single Page Allocator (One page at a time)
    - Multi Page Allocator (Set of pages at a time)
    - Large Page Allocator (Large page size 16 MB)
    - Reserved Page Allocators (Pages reserved for emergency
2. Virtual Allocators (VirtualAlloc APIs)
3. Shared Memory Allocators (Windows File Mapping APIs)

DBCC MEMORYSTATUS
select * from sys.dm_os_memory_nodes

http://blogs.msdn.com/b/slavao/archive/2005/02/11/371063.aspx

SQLOS – Memory Manager Components

2. Memory Clerks

Memory nodes are hidden from memory manager users.
If a client of memory manager needs to allocate memory
it first creates a memory clerk.

Types: Generic, Cache store, User store and Object store

sys.dm_os_memory_clerks - to track and control amount of memory consumed by a memory component

Select * from sys.dm_os_memory_clerks

http://blogs.msdn.com/b/slavao/archive/2005/02/11/371063.aspx

SQLOS – Memory Manager Components

3. Memory Objects

SQL Server components use memory objects instead of memory clerks. Memory objects use the page allocator interface of the memory clerk to allocate pages.

```
select * from sys.dm_os_memory_objects
```

SQLOS – Memory Management

Demo

Questions?