64-bit Solutions on SQL Server 2005

Dean Bethke
Technology Architect
Microsoft Technology Team

Agenda

- 64-bit Platforms
- Benefits of 64-bit SQL
- 64-Bit Challenges and Best Practices
  - Memory Configuration
  - NUMA
  - Performance

64-bit Platforms

- The mainstream high-volume server platforms are now 64-bit OS ready
- Your choice is whether to run 64 bit SQL on these platforms, or to run 32-bit SQL on the 64-bit or 32-bit operating system
- For high-scale 64-bit systems, you also need to choose between Itanium (IA64) and x64 offerings

Properties of Itanium (IA64)

- Runs 64-bit Windows, drivers and software specifically compiled for the Itanium instruction set
- Runs 32-bit software without being recompiled
  - New 32-bit execution layer http://www.microsoft.com/windowsserver2003/64bit/ipf/ia32el.mspx
- Cannot act like an x86 processor or boot 32-bit Windows
- Does not run versions of Windows or 64-bit drivers compiled for x64

Intel Itanium-2 offerings

- 4 socket and above servers (→ 64 sockets)
  - 1.6Ghz and 9MB L3 cache (at 6.4 gigabits/sec system bus bandwidth)
  - Recent bump to 1.66Ghz but faster 667Mhz FSB (at 10.6 gigabits/sec)
- Dual-core now available
  - Montecito
    - Dual-core, huge cache (24MB L3), but still 1.66 Ghz
  - Montvale+
    - Multi-core and massive cache, slated 2007
  - Massive scalability database target market
- Main Vendors
  - HP, Unisys, NEC, Fujitsu

Properties of x64

- Runs 64-bit Windows, drivers and software specifically compiled for X64 instruction set
- Can act like an x86 processor when an X64 system is booted into a 32-bit operating system and as such runs all 32-bit versions of Windows commercially available today
- Runs 32-bit software without recompilation
- Does not run Itanium versions of Windows nor drivers compiled for Itanium

x64 Offerings

- Two chip vendors, same OS required
  - AMD (Opteron)
    - Eliminates FSB and uses HyperTransport
    - Currently limited to 1MB L2 cache
  - Intel (Xeon EM64T)
    - Large cache, NetBurst & Hyperthreading
    - Features 4MB and 8MB L3 cache versions
- Generally aimed at <= 4-socket, however…
  - Unisys - 32 socket (8 socket interconnect)
  - IBM – 16 socket (4 socket interconnect)
  - Sun – 8 socket (AMD)
- HP – offers both AMD & Intel (max 4-socket)
- Dell – currently offers Intel for Dell servers

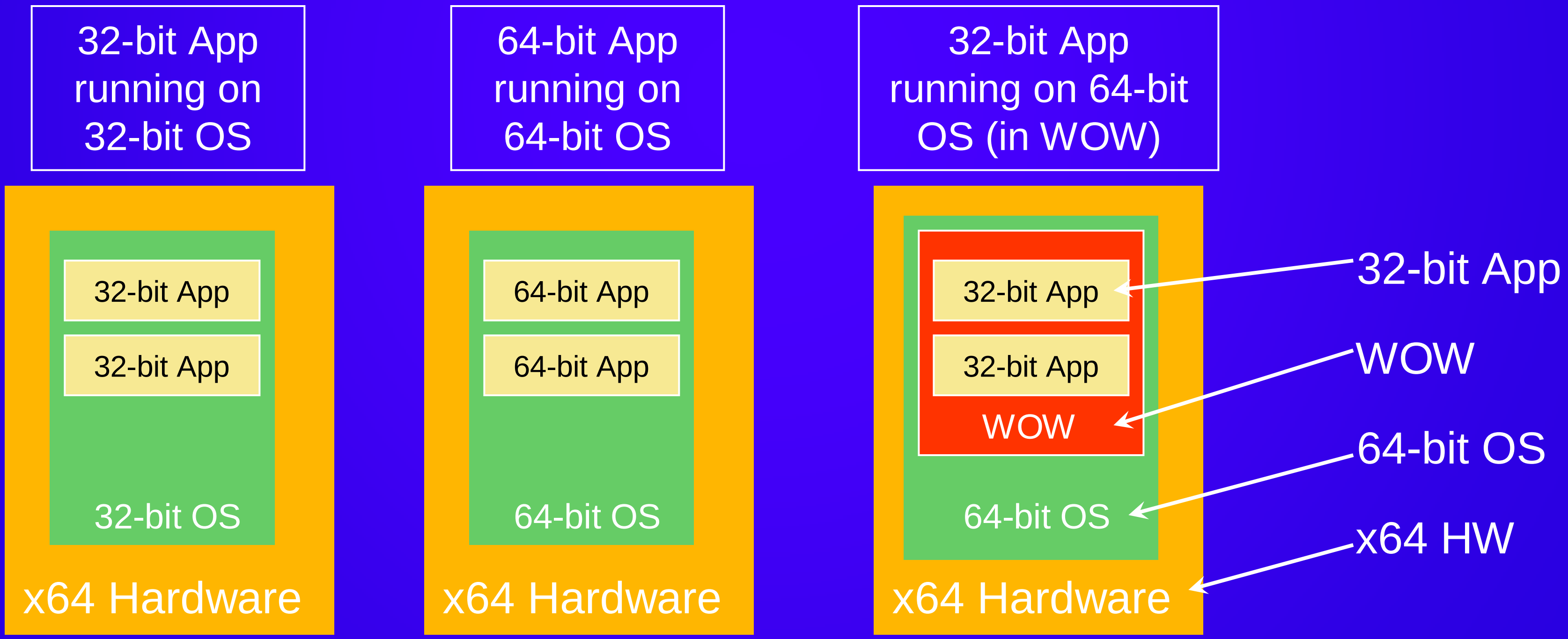
Multicore and Hyperthreading

- Multicore chips scale very effectively
  - Dual-core offers >> 50% performance benefit relative to single core
  - SQL Server’s operates as though each core is a separate CPU
    - Separate SQL Scheduler assigned to each core
  - But SQL is priced per socket, not per-core
- Hyperthreading does *not* benefit typical SQL workloads
  - Recommend disabling Hyperthreading
  - Potentially overloads a single core with multiple concurrent scheduler tasks
  - Multiple threads can thrash the CPU cache

Windows on Windows (WOW)

What is it?

- WOW allows 32-bit applications to run on 64-bit Operating Systems
- **SQL Server will NOT support both 32-bit SQL in WOW and native 64-bit SQL co-existing on same server**
  - Example: Once you install 32-bit SQL Server Engine the server **Cannot** run 64-bit Engine on same hardware



x64 Hardware = AMD64 and Intel EM64T

What is 64-bit SQL Server 2005?

- One code base for all platforms of SQL Server
- Flat memory addressing
- IA64 and X64
- Data files compatible with 32-bit SQL
  - Easy Database Migration & Integration
    - Detach/Attach, Log shipping, Replication etc.
- Massive Scale-up support
  - 64-way HP*, 32-way NEC, 32-way Unisys
  - >1,000,000 TPC number*
- 8-node Fail-over clustering support

64-bit gaps in SQL 2005

- VS not supported on Itanium
  - Although management tools are supported
- So BIDS not supported on Itanium – affects SSIS
  - Must develop and debug packages on a 32-bit server and deploy to 64-bit
  - Watch for driver compatibility
- x64 BIDS runs in the WOW
  - SSIS needs 32-bit drivers at pipeline construction time and 64-bit drivers at execution time
    - Some x64 drivers for external RDBMS are still problematic

32-bit SQL Server on x64

- 32-bit SQL Server 2005 (server & tools) supported on x64 running 64-bit Windows
  - Under WOW64, SQL Server can access FULL 4GB of RAM, as well as AWE (no need to enable /3GB)
    - Only without CLR enabled; with CLR enabled only 3GB
- Why would you want to run 32-bit OS instead?
  - Drivers
  - Coexistence with other applications or tools that aren’t WOW-certified

SQL Server on X64, Itanium

SQL Server Edition	x64			Itanium	
	32-bit OS	64-bit OS WOW64	64-bit OS	64-bit OS WOW64	64-bit OS
SQL Server 2000 32-bit (SP4)	Yes	Yes	No	No	No
SQL Server 2000 64-bit Itanium	No	No	No	No	Yes
SQL Server 2005 32-bit	Yes	Yes	No	No	No
SQL Server 2005 64-bit x64*	No	No	Yes	No	No
SQL Server 2005 64-bit Itanium*	No	No	No	No	Yes
SQL Server Express 32-bit	Yes	Yes	No	No	No
SQL Server Mgmt Studio 32-bit	Yes	Yes	No	Yes	No
SQL Server BI Dev Studio 32-bit	Yes	Yes	No	No	No

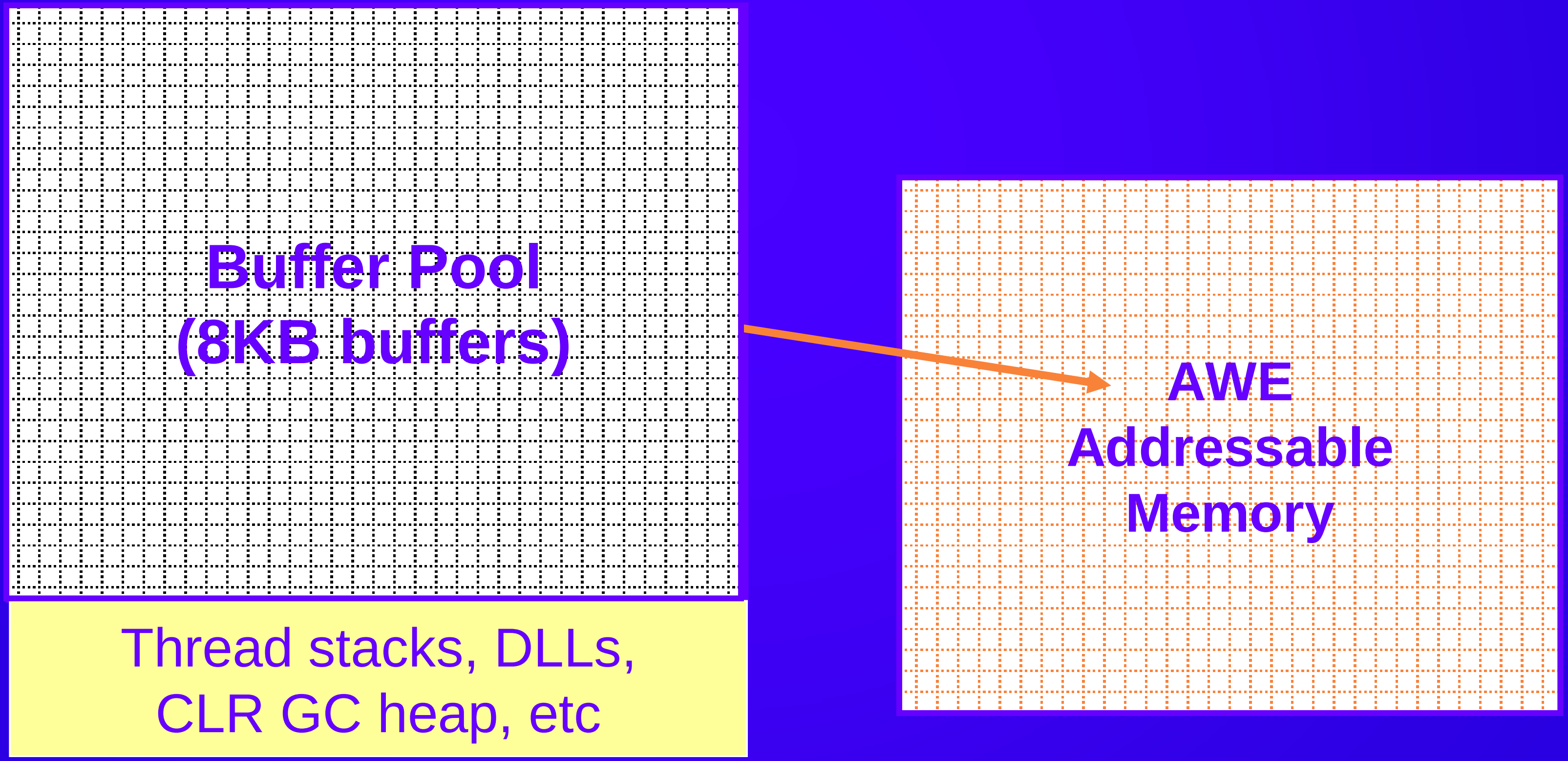
Agenda

- 64-bit Platforms
- Benefits of 64-bit SQL
- 64-Bit Challenges and Best Practices
  - Memory Configuration
  - NUMA
  - Performance

Memory Primer

- 32-bit applications
  - All 32-bit applications are limited to a 4GB virtual address space (VAS)
  - An application can utilize 2GB (3GB using /3gb, or 4GB in x64 WOW). Remainder is reserved for OS addresses
  - Using AWE, some applications can use larger amounts of memory by mapping / unmapping additional memory into the VAS
- 64-bit applications
  - Flat, huge virtual address space
  - No mapping needed for large memory access

SQL Server Memory



Buffer Pool
(8KB buffers)

Thread stacks, DLLs,
CLR GC heap, etc

AWE
Addressable
Memory

Where Will 64-Bit SQL Help?

Relational Server

- Memory hungry workloads – aside from DB buffer pages – consuming:
  - Plan Cache: Lots of stored procedures and dynamic SQL batches (OLTP)
  - Workspace Memory: Large number of concurrent hash joins or aggregates; large-scale sorts and index builds (Data Warehouse)
  - Connection memory: Large numbers of connections
  - Thread memory: High Concurrency OLTP
  - CLR GC Heap memory: CLR SP allocations
  - Lock Memory: Large scale OLTP

All are common scenarios for scaled-up OLTP and Data Warehouse applications, utilizing many CPUs

Identifying Memory Pressure

Relational Server

- Workspace Memory
  - Queries queued waiting on Memory Grants
  - With large percentage of VM devoted to workspace memory
- Plan Cache
  - Large number of compiles/sec
  - Low cache hit ratio
- Locks
  - Large percentage of VM occupied by locks

Where Will 64-Bit SQL Help?

Analysis Services

- Dimensions
  - AS 2005 queries perform much better when dimension members can be cached in memory
  - Individual sets of related attributes must fit in memory during dimension processing
  - Key attributes for at least the largest dimension must fit in memory during partition processing
- Processing Buffers and Data Cache
- Windows Filesystem Cache
  - Can grow very large on 64-bit and boost performance accessing the files containing aggregates and measures
- NOTE: No AWE support in Analysis Services

Identifying Memory Pressure

Analysis Services

- Inadequate process buffers in AS2005
  - Processing becomes single threaded during aggregation phase
    - CPU utilization very low during partition processing
    - Temp files used (monitor in Filemon or Perfmon counters)
  - Indicates need to increase Process Buffer Size
- Pressure on AS Data Cache
  - Perfmon: Agg Cache Hit Ratio, Evictions/sec

Where Will 64-Bit SQL Help?

Integration Services 2005

- Specific large-scale transformations can consume a lot of virtual memory:
  - Sort
  - Aggregate
  - Key Lookup (cached)
- Packages may fail on 32-bit when 3GB is insufficient virtual address space to process large streams using the above operations

Where Will 64-Bit SQL Help?

Scale-Up and Performance

- Itanium platforms are today’s choice for workloads requiring > 8 CPUs
    - Very large Data Warehouses
    - Massive OLTP
    - Itanium offers excellent scaling characteristics, multi-node bus architecture
    - X64 Xeon scaled-up servers have just arrived
- X64 offers fastest CPU performance today
    - Single-threaded performance on X64 can surpass Itanium today

Agenda

- 64-bit Platforms
- Benefits of 64-bit SQL
- 64-Bit Challenges and Best Practices
  - Memory Configuration
  - NUMA
  - Performance

64-Bit Challenges

- 64-bit drivers for Data Access
  - Analysis Services and Integration Services require the 64-bit versions of 3rd party OleDB drivers to support data sources such as Oracle, Informix, etc.
  - No 64-bit ODBC access supported
  - Verify before purchasing your server!
- Memory Configuration of Components
  - Multiple SQL Instances on server
  - Multiple Service components on server, e.g. AS, SQL Relational, SSIS
  - Both scenarios were simpler on 32-bit – memory was isolated between apps

Memory Configuration Alternatives

- Use *separate* hardware partitions or *separate* servers for solutions that combine Relational, AS and SSIS
  - Eliminates risks of cross-application memory contention
  - But this may underutilize CPU resources

OR

- Manage memory use *explicitly* for each component running on a *single* server

64-bit Memory Configuration
Relational Server

- If multiple SQL Server instances share the server
  OR
- If multiple applications (AS, SSIS) coexist on the server
  - Best Practice is to place a memory ceiling each SQL instance
    - Or even use a fixed memory size
  - Lock pages in memory to prevent OS paging of SQL under memory pressure

64-bit Memory Configuration
Relational Server

- Min, Max Server Memory determines how dynamic SQL’s Buffer Pool memory is
  - More Dynamic:
    - + Allows SQL to balance memory consumption against demands of other applications
    - - Reduces predictability of performance for critical memory-dependent workloads
    - - SQL Performance problems likely when there are sudden external memory demands
  - Less dynamic
    - + Guarantees resources for a critical workload
    - - May leave insufficient memory for other applications
    - - May leave insufficient memory for non-buffer pool components (CLR GC heap, DLLs, etc)

64-bit Memory Configuration
Relational Server

- To limit SQL RDBMS memory consumption
  - Determine desired size of total RDBMS memory footprint = Msql
    - Ensure that
      Msql < Total Server Memory – ~0.5GB for OS
      – Max memory required by other apps or instances
  - Reserve enough room outside of buffer pool for
    - SQL Threads
      - Threads will consume 4MB each on Itanium, 2MB on X64
        - (= 1GB for 256 Worker Threads on Itanium!)
    - XPs, In-Proc OleDB drivers, CLR GC heap
  - Calculate & set the resulting size of buffer pool desired
    - Bpool = Msql - Threads - XPs - CLR GC heap - etc
    - Sp_configure ‘Max Server Memory’ = Bpool
    - Consider: Sp_configure ‘Min Server Memory’ = Bpool
  - NOTE: Memory outside of Buffer Pool cannot be limited

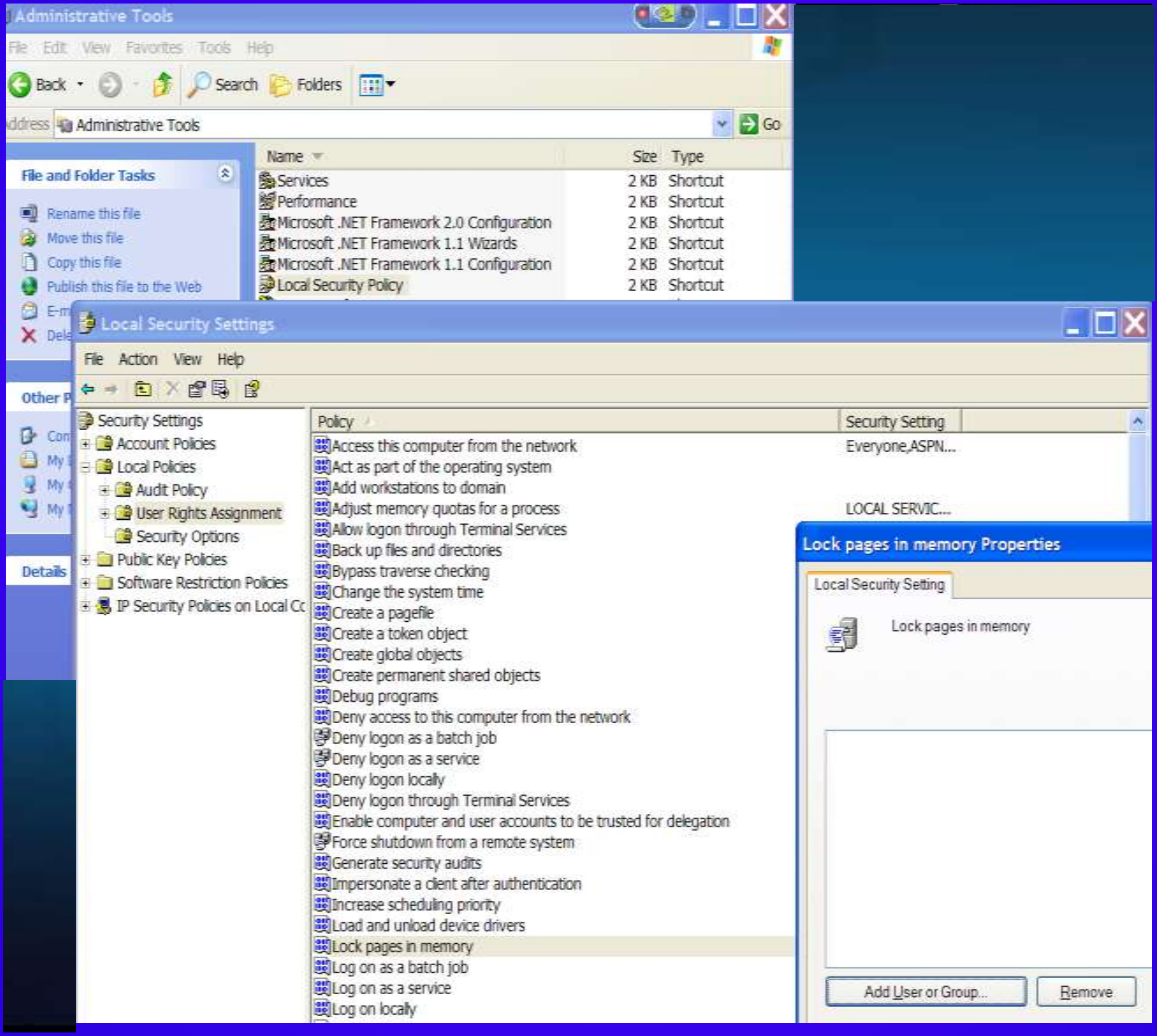
64-bit Memory Configuration
Relational Server

- SQL responds to external memory pressure
  - If memory is dynamic (Max <> Min) SQL will try to reduce its working set when notified by OS
    - Sometimes SQL doesn’t respond fast enough and the OS will page out part of SQL’s address space
  - If memory is fixed (Max = Min), the OS may page out part of SQL’s address space under memory pressure
- Either way, OS paging of SQL is very bad
  - System becomes unresponsive
  - Query times can become impossibly long
  - Launching an unexpected new process or growing an existing one can cause OS to page SQL
- OS Paging of *other* apps on the server is disk and CPU intensive
  - Can also degrade SQL performance

64-bit Memory Configuration
Relational Server

- Prevent OS paging of Buffer Pool by Locking Pages in Memory
  - Automatic if SQL Service Account has Lock Pages in Memory privilege
    - Control Panel / Admin Tools / Local Security Policy
- SQL Memory can still be dynamic
- Note: Other parts of SQL may still be paged, along with other applications

Lock Pages Permission



Address: Administrative Tools

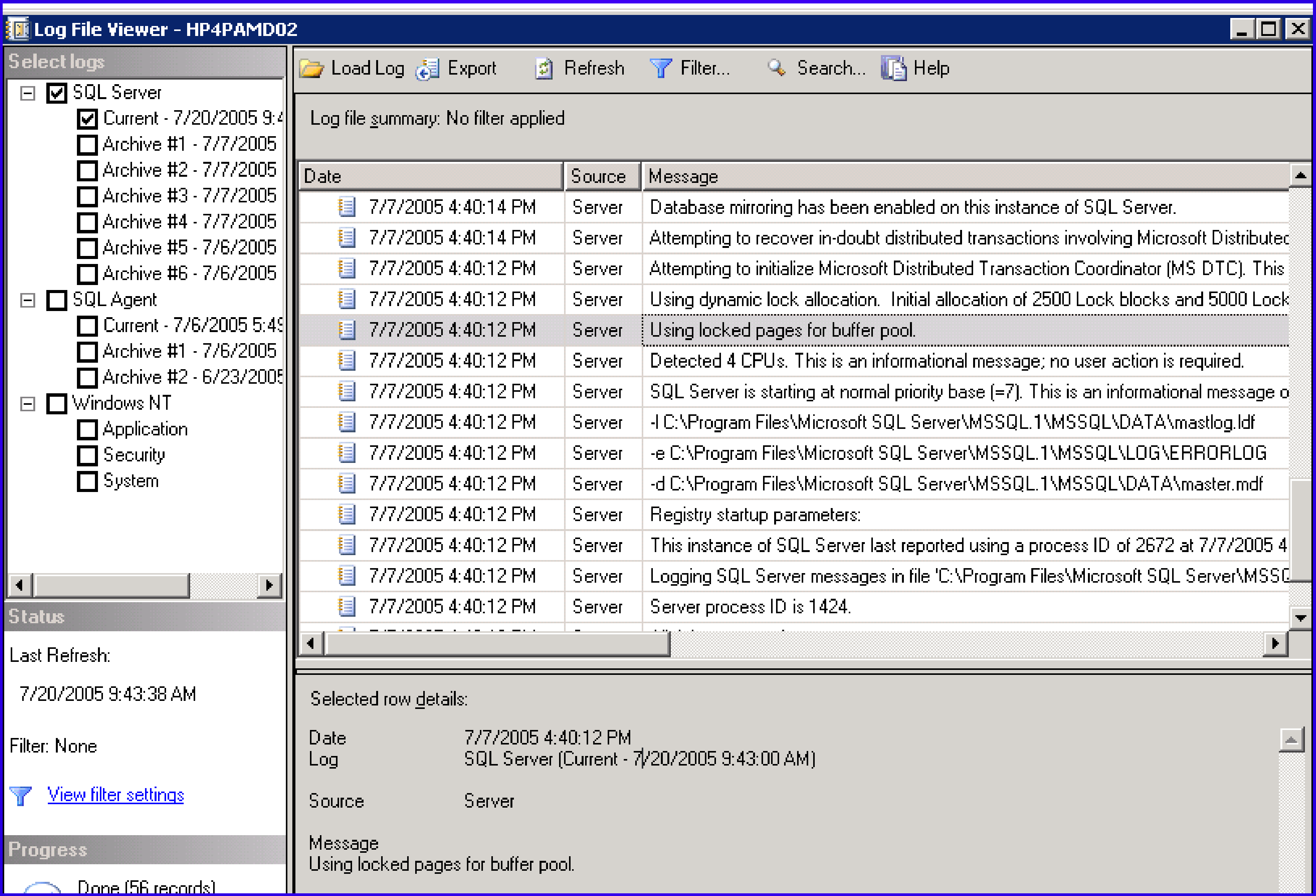
Name	Size	Type
Services	2 KB	Shortcut
Performance	2 KB	Shortcut
Microsoft .NET Framework 2.0 Configuration	2 KB	Shortcut
Microsoft .NET Framework 1.1 Wizards	2 KB	Shortcut
Microsoft .NET Framework 1.1 Configuration	2 KB	Shortcut
Local Security Policy	2 KB	Shortcut

- Security Settings
  - Account Policies
  - Local Policies
    - Audit Policy
    - User Rights Assignment
    - Security Options
  - Public Key Policies
  - Software Restriction Policies
  - IP Security Policies on Local Cc

Policy	Security Setting
Access this computer from the network	Everyone,ASPN...
Act as part of the operating system	
Add workstations to domain	
Adjust memory quotas for a process	LOCAL SERVIC...
Allow logon through Terminal Services	
Back up files and directories	
Bypass traverse checking	
Change the system time	
Create a pagefile	
Create a token object	
Create global objects	
Create permanent shared objects	
Debug programs	
Deny access to this computer from the network	
Deny logon as a batch job	
Deny logon as a service	
Deny logon locally	
Deny logon through Terminal Services	
Enable computer and user accounts to be trusted for delegation	
Force shutdown from a remote system	
Generate security audits	
Impersonate a client after authentication	
Increase scheduling priority	
Load and unload device drivers	
Lock pages in memory	
Log on as a batch job	
Log on as a service	
Log on locally	

Lock pages in memory

Verifying Locked Pages



Select logs

- ☑ SQL Server
  - ☑ Current - 7/20/2005 9:4
  - □ Archive #1 - 7/7/2005
  - □ Archive #2 - 7/7/2005
  - □ Archive #3 - 7/7/2005
  - □ Archive #4 - 7/7/2005
  - □ Archive #5 - 7/6/2005
  - □ Archive #6 - 7/6/2005
- □ SQL Agent
  - □ Current - 7/6/2005 5:49
  - □ Archive #1 - 7/6/2005
  - □ Archive #2 - 6/23/2005
- □ Windows NT
  - □ Application
  - □ Security
  - □ System

Status

Last Refresh:

7/20/2005 9:43:38 AM

Filter: None

View filter settings

Progress

Done (56 records)

Log file summary: No filter applied

Date	Source	Message
7/7/2005 4:40:14 PM	Server	Database mirroring has been enabled on this instance of SQL Server.
7/7/2005 4:40:14 PM	Server	Attempting to recover in-doubt distributed transactions involving Microsoft Distributed
7/7/2005 4:40:12 PM	Server	Attempting to initialize Microsoft Distributed Transaction Coordinator (MS DTC). This
7/7/2005 4:40:12 PM	Server	Using dynamic lock allocation. Initial allocation of 2500 Lock blocks and 5000 Lock
7/7/2005 4:40:12 PM	Server	Using locked pages for buffer pool.
7/7/2005 4:40:12 PM	Server	Detected 4 CPUs. This is an informational message; no user action is required.
7/7/2005 4:40:12 PM	Server	SQL Server is starting at normal priority base (=7). This is an informational message o
7/7/2005 4:40:12 PM	Server	-l C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\mastlog.ldf
7/7/2005 4:40:12 PM	Server	-e C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\LOG\ERRORLOG
7/7/2005 4:40:12 PM	Server	-d C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\master.mdf
7/7/2005 4:40:12 PM	Server	Registry startup parameters:
7/7/2005 4:40:12 PM	Server	This instance of SQL Server last reported using a process ID of 2672 at 7/7/2005 4
7/7/2005 4:40:12 PM	Server	Logging SQL Server messages in file 'C:\Program Files\Microsoft SQL Server\MSSQ
7/7/2005 4:40:12 PM	Server	Server process ID is 1424.

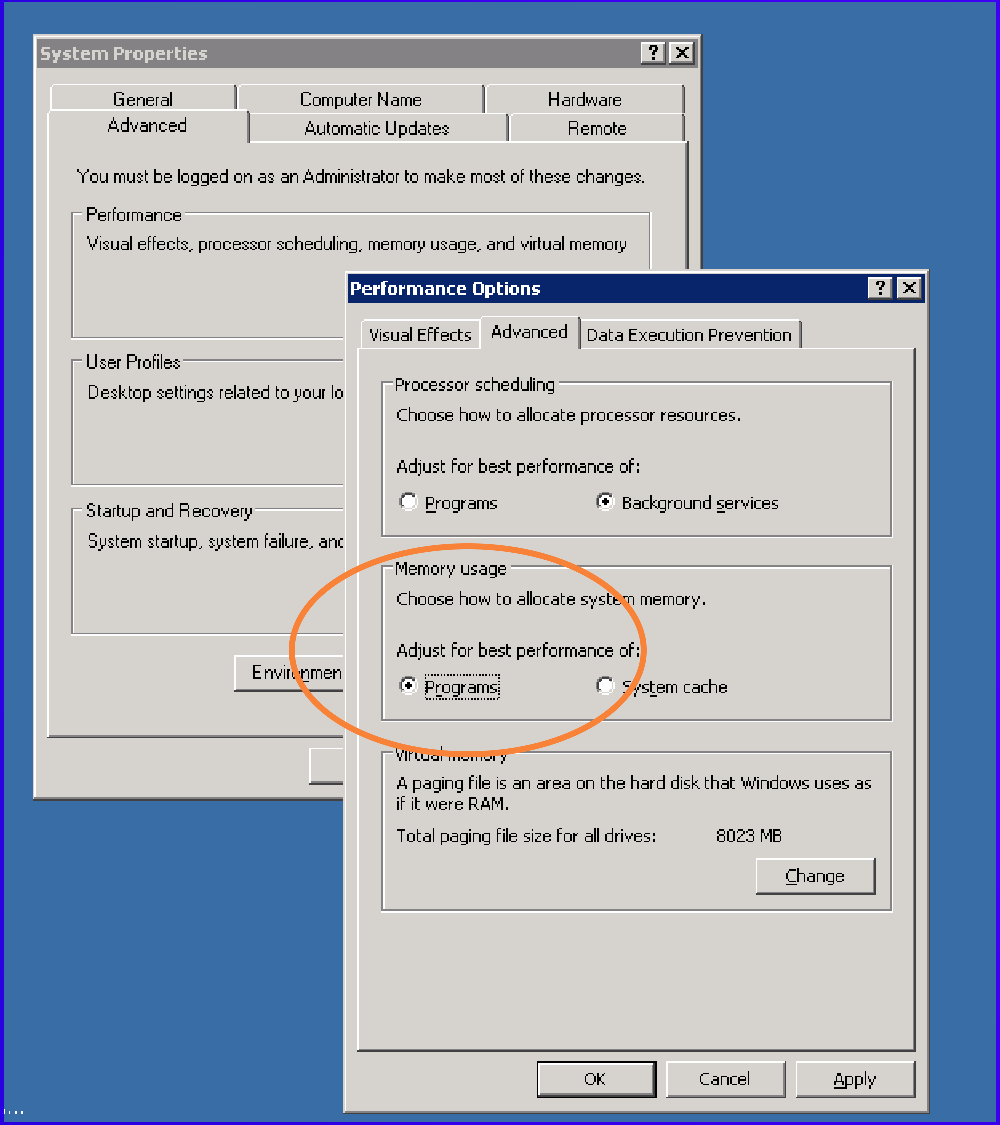
Selected row details:

Date 7/7/2005 4:40:12 PM
Log SQL Server (Current - 7/20/2005 9:43:00 AM)

Source Server

Message
Using locked pages for buffer pool.

Minimize Filesystem Cache



System Properties

General | Computer Name | Hardware
Advanced | Automatic Updates | Remote

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your lo

Startup and Recovery
System startup, system failure, anc

Performance Options

Visual Effects | Advanced | Data Execution Prevention

Processor scheduling
Choose how to allocate processor resources.

Adjust for best performance of:
○ Programs ● Background services

Memory usage
Choose how to allocate system memory.

Adjust for best performance of:
● Programs ○ System cache

Virtual memory
A paging file is an area on the hard disk that Windows uses as if it were RAM.

Total paging file size for all drives: 8023 MB

- Potential for enormous filesystem cache on 64-bit can create external memory pressure
- Configure OS to favor memory for programs, not filesystem

SQL Memory Acquisition

- By default SQL engine will *not* acquire all requested memory upon startup
  - Grows buffer pool gradually as it needs the memory
- If another application on the server grabs memory first, SQL may not be able to obtain all the memory you have configured it to use
  - Rare cases where you want to start SQL first & run a workload to acquire memory prior to launching other memory-consuming apps

64-bit Memory Configuration
Analysis Services

- Consumes memory in two ways
  - Virtual Memory for Analysis Services Service
    - Bounded by configuration settings
  - OS Memory used as Filesystem cache
    - Potentially unbounded

- If AS is running on a server with other critical services (SQL or SSIS):
  - Best Practice to establish a maximum memory size for Analysis Services process *and* limit growth of Filesystem Cache

64-bit Memory Configuration
Analysis Services

- To limit Analysis Services memory consumption, set

  Analysis Services Properties: Advanced
  or Config File

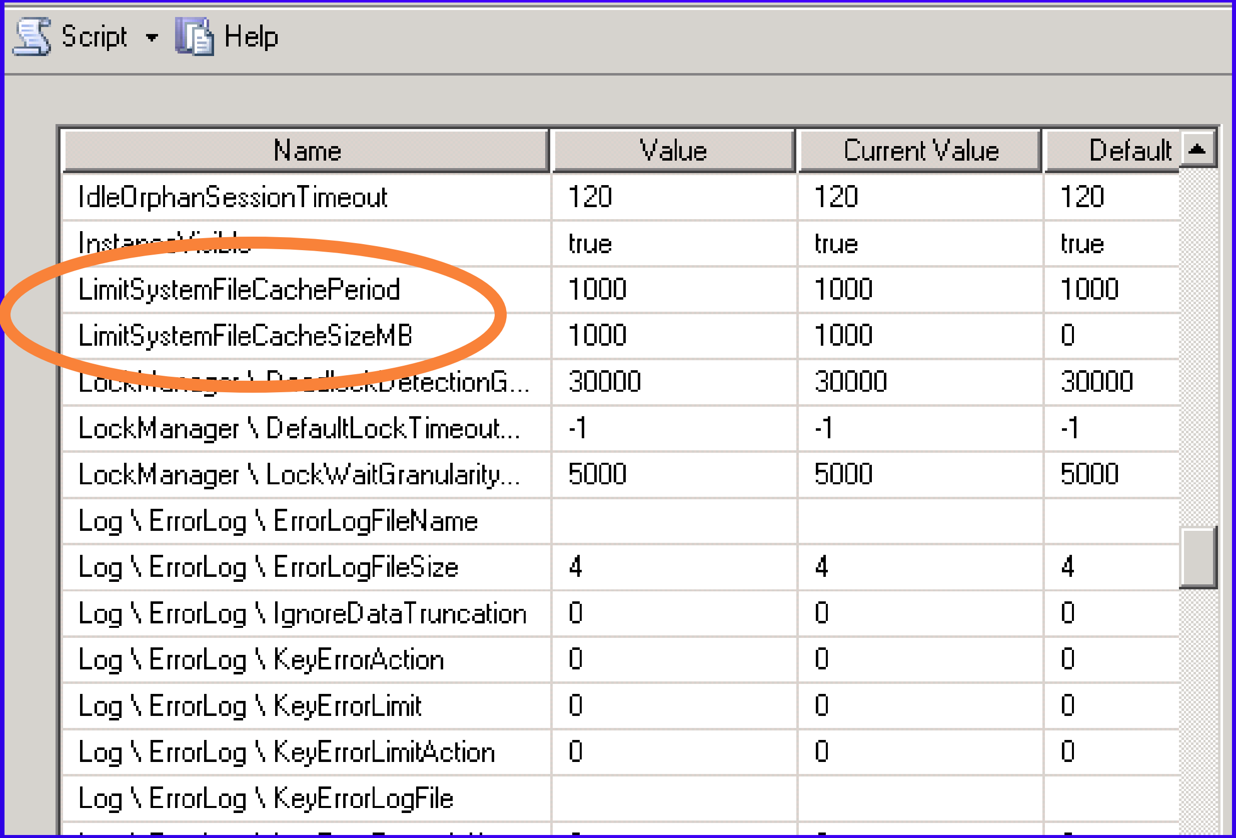
  Memory \ TotalMemoryLimit

  - No mechanism to Lock Pages in Memory prior to SP2
  - SP2 can pre-allocate memory with <PreAllocate> config entry

64-bit Memory Configuration
Analysis Services

- Filesystem Cache – filled by AS buffered IO
  - Can grow *without limit* and even page out other applications
    - Was limited to ~1GB max on 32-bit
    - If SQL has Locked Pages, Relational Server will not be paged out as cache grows
    - However, Analysis Services process and other components can page out if this grows too large
  - System Cache Trimming
    - Built-in to AS2005, but off by default
    - Set Advanced Server Properties in SSMS or msmdsrv.ini
      - LimitSystemFileCacheSizeMB (MB)
      - LimitSystemFileCacheSizePeriod (Milliseconds)
    - Takes immediate effect without a restart

AS2005 Properties to Limit Filesystem Cache

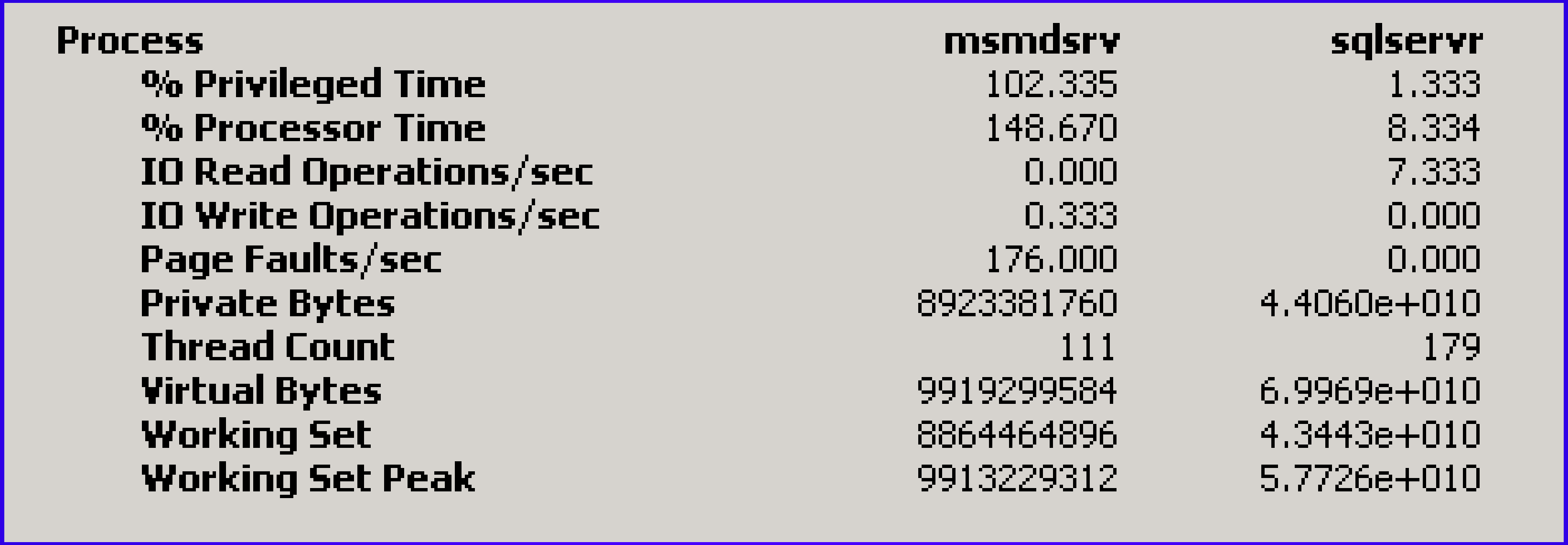


Name	Value	Current Value	Default
IdleOrphanSessionTimeout	120	120	120
InstanceVisible	true	true	true
LimitSystemFileCachePeriod	1000	1000	1000
LimitSystemFileCacheSizeMB	1000	1000	0
LockManager \ DeadlockDetectionG...	30000	30000	30000
LockManager \ DefaultLockTimeout...	-1	-1	-1
LockManager \ LockWaitGranularity...	5000	5000	5000
Log \ ErrorLog \ ErrorLogFileName			
Log \ ErrorLog \ ErrorLogFileSize	4	4	4
Log \ ErrorLog \ IgnoreDataTruncation	0	0	0
Log \ ErrorLog \ KeyErrorAction	0	0	0
Log \ ErrorLog \ KeyErrorLimit	0	0	0
Log \ ErrorLog \ KeyErrorLimitAction	0	0	0
Log \ ErrorLog \ KeyErrorLogFile			

64-bit Memory Configuration
Integration Services

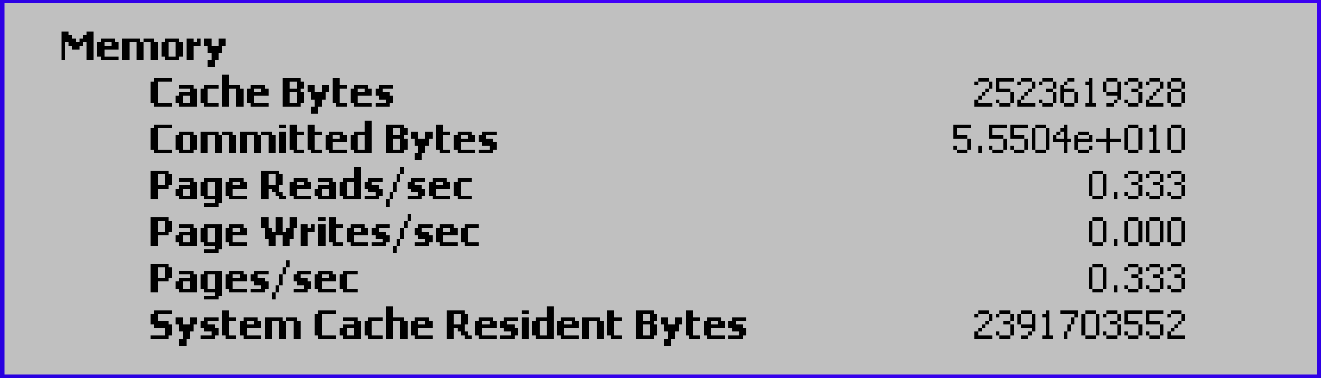
- You cannot impose explicit memory limits within SSIS
- Simple pipelines or control packages will not be significant consumers of memory
- *Large* Sort, Aggregation and Cached Lookup operations in a pipeline can consume a lot of virtual memory
    - Hint: Retrieve *minimum* necessary fields for lookup operations
    - If these create too much memory pressure, consider using DB operations instead, or a separate server / partition

Key Memory Counters



```
Process                                  msmdsrv          sqlservr
        % Privileged Time                102.335             1.333
        % Processor Time                 148.670             8.334
        IO Read Operations/sec             0.000             7.333
        IO Write Operations/sec            0.333             0.000
        Page Faults/sec                  176.000             0.000
        Private Bytes                 8923381760       4.4060e+010
        Thread Count                         111               179
        Virtual Bytes                 9919299584       6.9969e+010
        Working Set                   8864464896       4.3443e+010
        Working Set Peak              9913229312       5.7726e+010
```

NOTE: sqlservr Private Bytes and Working Set are *not correct* in Perfmon when *Lock Pages* are used.



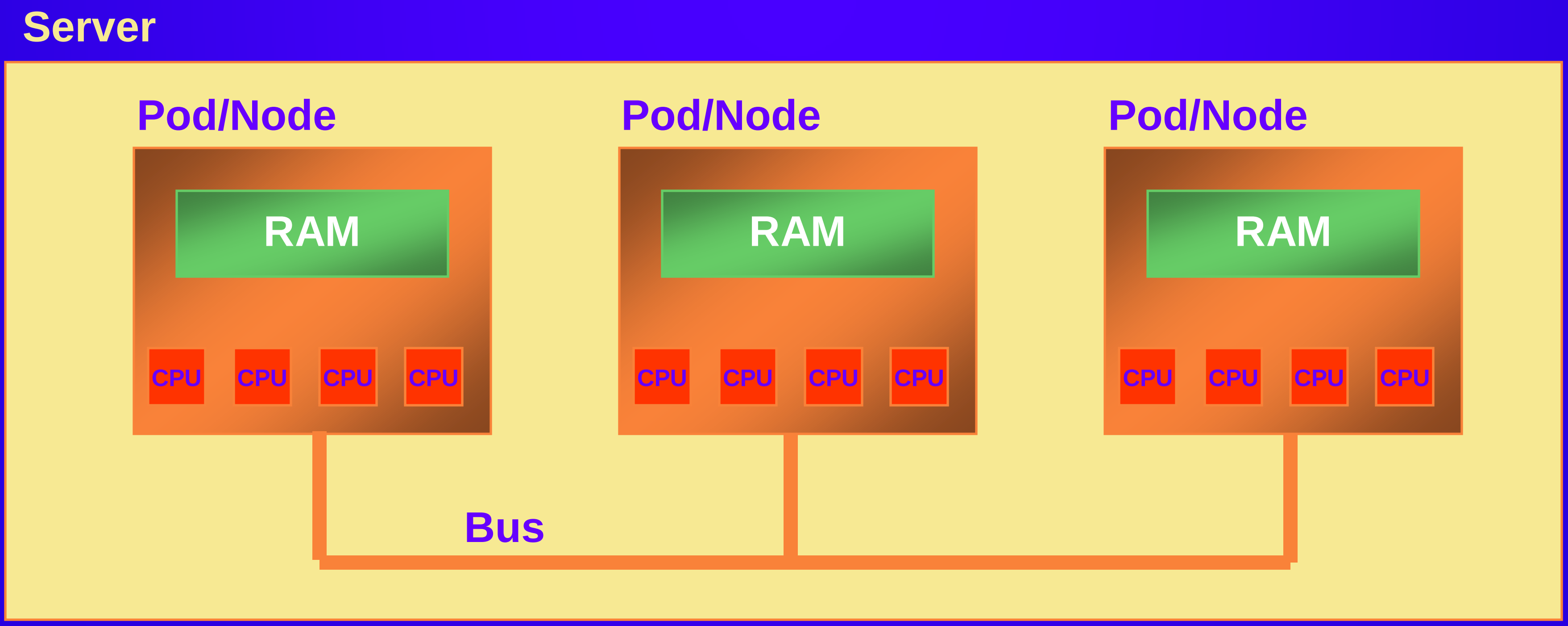
```
Memory
        Cache Bytes                           2523619328
        Committed Bytes                      5.5504e+010
        Page Reads/sec                             0.333
        Page Writes/sec                            0.000
        Pages/sec                                  0.333
        System Cache Resident Bytes           2391703552
```

Agenda

- 64-bit Platforms
- Benefits of 64-bit SQL
- 64-Bit Challenges and Best Practices
  - Memory Configuration
  - NUMA
  - Performance

NUMA Considerations

- All 64-bit servers that scale-up are NUMA platforms
  - Non Uniform Memory Access
  - Longer latencies for accessing memory on remote nodes



NUMA Servers

- All NUMA servers can be configured in BIOS to expose to the OS either
  - ‘NUMA Memory’, or
  - ‘Interleaved Memory’
- NUMA memory – OS associates a specific memory address range with each pod
- Interleaved memory – memory addresses are spread, fine grained, across all pods

SQL2005 and NUMA

- NUMA memory configuration is the preferred choice on SQL2005 servers with > 4 cores
  - Some OLTP workloads are 60% faster when using NUMA instead of Interleaved memory
- A SQL connection will remain on the same NUMA node for its lifetime
  - Potential for leveraging local cache for all of its tasks
  - Can direct specific connections to specific nodes

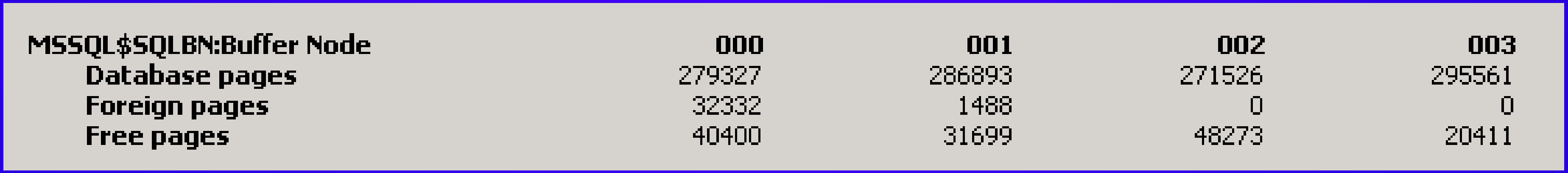
SS2005 RDBMS

NUMA Memory Management

- SQL RDBMS instantiates an abstraction for each NUMA node on the server
  - A ‘SQL Node’ including lazywriter, bpool, schedulers, IO port, etc.
- Each SQL Node obtains memory from Windows, proportionally
  - e.g. memory per node = SQL memory/#Nodes
- Windows will *try* to allocate to a SQL Node memory to a that is local to its NUMA node
  - If other processes have grabbed local memory, the SQL Node will have ‘foreign pages’ allocated

NUMA Memory
Management Continued

- SQL Tasks running on a node will use pages *allocated* to that node (local or foreign) when possible
    - But will use another node’s pages if DB pages are already cached there
    - Will use another node’s pages to satisfy a memory grant when necessary
- Node allocation of ‘foreign pages’ can be monitored in Perfmon:



MSSQL$SQLBN:Buffer Node	000	001	002	003
Database pages	279327	286893	271526	295561
Foreign pages	32332	1488	0	0
Free pages	40400	31699	48273	20411

Some NUMA Gotchas

- AS and SSIS are mostly *not* NUMA aware
  - Pre-SP2
    - No guarantees of evenly distributed memory allocation
    - OS will tend to fill up a single node’s CPUs with work before exercising other CPUs
    - Common to see only 1 busy set of nodes and remainder of server idle until a node saturates
  - SP2 Improvements
    - SP2 has NUMA improvements for processing
    - SP2 can pre-allocate memory per NUMA node with config entry
- Consider starting up SQL Server and warming cache before launching AS
  - May be effective in very large cube environments
  - Avoids AS potentially grabbing all local memory for one node
  - Avoids filesystem cache from stealing pages disproportionately from a single node – which could create a SQL node starved of local pages

Agenda

- 64-bit Platforms
- Benefits of 64-bit SQL
- 64-Bit Challenges and Best Practices
    - Memory Configuration
    - NUMA
    - Performance

Performance Considerations

- SQL Server 64-bit working set is larger than 32-bit SQL
  - Long pointers
  - Code path less likely to be found in processor cache than equivalent 32-bit code
- Result: Apps that are *not* memory constrained on 32-bit (especially OLTP) may run ~10% *less efficiently* using 64-bit edition than 32-bit edition on the same machine
  - May observe more CPU cycles to do equivalent work
  - CPUs spin waiting for non-cached memory access

Performance Considerations

- If you are not under memory pressure, and do not need to scale up, and you want the absolutely fastest single-thread performance
  - Use X64 instead of Itanium
  - Consider using 32-bit SQL in the WOW or on 32-bit OS
  - Xeon’s larger L3 cache may have a noticeable impact relative to Opteron
- If you need to scale up for greater throughput
  - Use Itanium with large number of CPUs

    or
  - Consider newest scaled-up X64 servers by Unisys

Summary: 64-bit ‘gotchas’

- Multiple components competing for memory
- Potential NUMA effects
- Potential performance penalty due to larger working set
- Driver availability
- Management tool support

Resources

- SQLCAT Team Blog: http://blogs.msdn.com/sqlcat
- SQL Server memory management, 64-bit and NUMA
  - Slava Oks Blog
    - http://blogs.msdn.com/slavao
- Analysis Services Configuration and Performance
  - Project REAL
    - http://www.microsoft.com/sql/bi/projectreal
- Integration Services
  - Blogs by Ashvini Sharma, Donald Farmer at sqljunkies.com
    - http://www.sqljunkies.com/WebLog/donald_farmer
    - http://sqljunkies.com/WebLog/ashvinis

Microsoft®

Your potential. Our passion.™

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.