Microsoft® Windows Server® 2008

Memory Provisioning Recommendations

Table of Contents

Overview

Physical memory is a critical component of all Windows Operating Systems and Servers, especially in environments which leverage the latest and most powerful Windows operating systems, server applications, and developer tools, such as Windows Server® 2008, SQL Server® 2008, and Visual Studio® 2008, respectively. Provisioning the optimal amount of physical memory for Windows operating systems and servers is more important today as servers with ever increasing numbers of high performance CPU cores become standard IT platforms. This paper defines guidelines and recommendations for determining the appropriate memory capacity for specific roles within Windows Server 2008, SQL Server 2008, and Visual Studio 2008.

This paper is intended for Microsoft and Kingston partners, resellers, and IT customers who plan to implement the latest and most powerful enabling Windows products in production scenarios and want to proactively plan for physical memory requirements.

Windows Memory Management

Microsoft's memory management technologies continue to evolve with each successive release of Windows Server software, and Windows Server 2008 is no exception. Many of the memory management changes are transparent outside of the operating system itself, so existing applications and drivers run without modification. Although the technical details of the actual memory management advances in Windows Server 2008 are beyond the scope of this paper, a brief discussion of Windows memory as it pertains to processor architecture is in order.

Table 1 lists the maximum memory capabilities of varying versions of Windows, based on the processor architecture.

		Windows Server 2003 RTM	Windows Server 2003 SP1	Windows Server 2003 R2	Windows Server 2003 SP2	Windows Server 2008
X86	Standard Edition	4 GB	4 GB	4 GB	4 GB	4 GB
	Enterprise Edition	32 GB	32 GB	64 GB	64 GB	64 GB
	Datacenter Edition	32 GB	32 GB	128 GB	128 GB	64 GB
X64	Standard Edition	--	--	--	--	32 GB
	Enterprise Edition	--	--	2 TB	2 TB	2 TB*
	Datacenter Edition	--	1 TB	2 TB	2 TB	2 TB*
IA64	Standard Edition	--	--	--	--	2 TB*
	Enterprise Edition	64 GB	1 TB	--	2 TB	
	Datacenter Edition	512 TB	1 TB	--	2 TB	

Table 1 – Windows Server Maximum Memory Configuration[1]

*** Note:** The 2 TB limit on Windows Server 2008 x64 is limited to hardware capabilities at the time of Microsoft certification. This number could increase as higher memory capacity systems become available. Also, IA64 version of Windows Server 2008 only comes in a single edition, "Windows Server 2008 for Itanium-Based Systems".

A major advancement in Windows Server 2008 is the support for x64 platforms with up to 32GB of RAM using *Standard Edition*. Previously this high RAM capacity was strictly reserved for *Enterprise* and *Datacenter Editions* of Windows Server 2003. With 64-bit architecture being supported in new servers, more organizations will take advantage of 64-bit platforms and leave behind the memory challenges associated with 32-bit architectures.

32-bit Windows Server Memory Management Challenges

By definition, 32-bit operating systems use 32 bits to identify memory locations in physical memory, resulting in a maximum of about 4.2 billion possible locations (2^{32} = 4,294,967,296, or 4GB). The Windows 32-bit architecture allocates this 4GB addressable virtual memory space to each running application, divided equally into two components; 2GB of virtual memory allocated to the individual application process and 2GB dedicated for kernel usage and shared between all running processes on the system.

[1] Data obtained from the Microsoft® web site.

The challenge arises from the types of information stored in this 2GB shared space, how the space is used by the operating system, and what occurs when the kernel memory space is exhausted. The shared 2GB kernel memory space is divided among the following components:

Non-Paged Pool – The non-paged pool contains kernel memory components that cannot be paged out of memory, such as device drivers and core operating system components.

Paged Pool – The paged pool area maintains the memory allocated from kernel components and drivers that can be paged out if necessary.

System Cache – The system cache maintains a map to memory locations containing files that are currently open by the system.

System Page Table Entries (PTEs) – System PTEs are reference tables contained in memory that map each application's process' virtual memory addresses to the physical memory address space in RAM. As more processes are running on a Windows Server, more PTEs are necessary to track the individual memory addresses.

In prior versions of Windows Server, the amount of memory allocated to each component is set by the operating system at boot up time and is fixed in size. Windows Server 2008 introduces an architectural change to the Windows Memory Manager, which enables dynamic kernel memory allocation of the above-listed kernel components. Instead of being fixed in size at boot up, as is the case with prior versions of Windows Server, the size of each can be dynamically adjusted to accommodate changing workload conditions.

Beginning with Windows Server 2003, all 32-bit Windows Server editions allow for a change in balance between user-mode and kernel-mode memory components by leveraging the /3GB switch in the boot.ini file. This allows each running application to be allocated a larger portion of the virtual memory space, 3GB, reducing the shared kernel virtual memory space to 1GB. This additional application virtual address space helps to reduce the amount of memory fragmentation in the virtual address space of many memory-intensive enterprise application services such as Microsoft Exchange and Microsoft SQL Server, but it still doesn't solve the problems associated with applications that require, and could benefit from, larger virtual memory address space.

64-bit Memory Architecture Advancements

The 64-bit architecture model provides huge advancements to the virtual memory model in the Windows Server operating system. 64-bit architecture now allows for over 18 trillion possible memory reference locations (2^{64} = 18,446,744,073,709,551,616, or 16 TB), substantially more than was possible in the 32-bit architecture. Table 2 – *x86 and x64 Kernel Memory Allocation Maximums* displays the changes in architecture from x86 to x64 memory allocation maximum values.

Architectural Component	X86-based Windows Server 2003	X64-based Windows Server 2003/2008
Kernel virtual address space	2 GB	8 TB
Virtual memory	4 GB	16 TB
Paged pool	470 MB	128 GB
Non-paged pool	256 MB	128 GB
System cache	1 GB	1 TB
System PTEs	660 MB	128 GB

Table 2 – x86 and x64 Kernel Memory Allocation Maximums[2]

As seen in the above table, the increased memory allocations maximums are staggering. In a fully-configured server, each running application on an x64 server is allocated up to 16TB of virtual memory space, with 8TB dedicated to each running application and 8TB shared among all kernel memory components, compared to 4GB, 2GB and 2GB, respectively on older 32-bit servers. The x64 architecture provides far more virtual memory space for large, memory-intensive applications, which in turn can lead to increased performance, as well as removing the 2GB shared kernel address space of traditional 32-bit architectures.

For a much more technical in-depth explanation of the advancements in memory management in Windows Server 2008, refer the Microsoft whitepaper "*Advances in Memory Management for Windows*", written by David Solomon and Mark Russinovich, available for download on Microsoft's web site (http://www.microsoft.com/whdc/system/cec/MemMgt.mspx).

Windows Server 2008 Roles and Memory Provisioning Guidelines

Server roles were introduced in Windows Server 2003, but are now fully realized in Windows Server 2008. Traditionally, Windows administrators were required to install individual components required for a particular purpose through Add/Remove Programs in the Windows Control Panel. Although Windows Server 2003 supported included role-based component installations, roles were not enforced and administrators could still install or remove individual services; this implementation created the possibility that unnecessary services could be arbitrarily enabled on a server with no mechanism for easy tracking or auditing of such actions. For instance, an administrator installing web services on a Windows 2003 Server could also install any number of other Windows components, such as DHCP or DNS services.

Windows Server 2008 now installs all services through the designation of server *roles*. Using the previous example, DNS or DHCP services must be loaded through the installation of the explicit roles associated with each. This new role-based installation process now enables organizations to easily identify what services are installed on Windows Server 2008 hosts without needing to perform network or system-level scans to discover individual active services.

[2] Table data from Microsoft Knowledgebase article KB294418, available on Microsoft's web site.

Terminal Server

Terminal Server has long maintained a high value proposition to enterprises in providing access to corporate applications and data to remote users without the need for costly and complex VPN solutions to extend the network edge to the user's remote location. In a Terminal Server deployment, applications are accessed via a remote desktop-redirection protocol where the data is safely maintained within the datacenter's secure perimeter.

Windows Server 2008 has advanced the feature set of Terminal Server to include functionality that was previously only available by leveraging a third-party add-on product, such as Citrix Presentation Server, Provision Networks, 2X or similar product. For example, the following features are now available as part of the Windows Server 2008 product offering:

> **Terminal Services RemoteApp** – A method to publish applications to terminal server users and have those applications behave as if they were running on the client's local system
>
> **Terminal Services Gateway** – A secure access method allowing users to connect to terminal servers over encrypted sessions using HTTPS
>
> **Terminal Services Web Access** – Users can access RemoteApp applications hosted on Windows terminal servers using only a web browser.
>
> **Terminal Services Session Broker** – User sessions can be load balanced across logical grouping of terminal servers in a farm
>
> **Terminal Services Easy Print** – A universal printing engine that enables driver-agnostic printing through terminal server sessions to client print devices

These new features will undoubtedly make Windows Server 2008 Terminal Server more enticing to a wider base of Microsoft users.

As anyone accustomed to designing and managing a Terminal Server deployment will attest, 64-bit architecture plays a large part in the value proposition of Windows Terminal Server, and coupled with the new features available in Windows Server 2008 many organization will be looking to leverage Terminal Server in their corporate infrastructure now more than ever.

In the past, 32-bit architectures created a major limitation in user density on individual terminal servers; however, Windows Server 2003 64-Bit Edition delivered huge gains in user density capabilities. Windows Server 2008 includes both 32-bit and 64-bit editions, but it is the 64-bit edition that draws the most attention for terminal server applications.

The 32-Bit Challenge

Anyone familiar with terminal server deployments most likely has run into challenges increasing user density per server. This is largely due to the limitations inherent in the 32-bit processing architecture discussed previously.

Of the types of information stored in the 2GB shared kernel virtual memory space, *paged pool*, *system cache* and *system page table entries (PTEs)* are often the limiting factors in 32-bit terminal server deployments. In the case of a terminal server potentially hosting hundreds of users, this shared kernel memory space may need to accommodate the kernel components of possibly thousands of processes. As the number of users and running application instances rises, the paged pool and system PTEs also rise, eventually to a point where all available memory allocated to either one is exhausted. Once kernel-shared memory is exhausted, the operating system can no longer spawn new processes and new user session requests are rejected while existing user sessions are prevented from launching any new applications.

As mentioned previously, Windows Server 2008 introduces an architectural change to the Windows Memory Manager, which enables dynamic kernel memory allocation of the above-listed shared virtual memory kernel components. This change can result in added user density of 10 to 20 concurrent sessions; however, this benefit to terminal server environments is marginal compared to the potential user densities on x64 editions of Windows Server 2008.

x64 Terminal Server Benefits

64-bit editions of Windows Server 2008 introduce new capabilities in Terminal Server user density, mainly due to the architectural benefits of the memory architecture in 64-bit implementations. When compared to the shared kernel memory components of x86 architecture, 64-bit Windows effectively removes the limitations imposed by 32-bit architecture. Table 3 shows an updated comparison of the two platforms and the equivalent memory allocation to each with the x86 shared kernel virtual memory space optimized for terminal server.

(Terminal Server optimized)

Architectural Component	X86-based Windows Server 2003	X64-based Windows Server 2003/2008
Kernel virtual address space	2 GB	8 TB
Virtual memory	4 GB	16 TB
Paged pool	260-480 MB	128 GB
Non-paged pool	256 MB	128 GB
System cache	1 GB	1 TB
System PTEs	~ 900 MB	128 GB

Table 3 - x86 and x64 Kernel Memory Allocation Maximums (Terminal Server Optimized)[3]

[3] Table information obtained from the Microsoft Whitepaper "*Terminal Server Scaling and Performance on x64-Based Versions of Windows Server 2003*", available for public download on Microsoft's web site

As observed in the table, the maximum System PTE allocation on a 64-bit platform is 128GB, enough to hold several thousand users' worth of application memory space translations. These architectural changes raise the theoretical user density to thousands of users per terminal server, although realistic totals will likely be governed by CPU, networking, and disk subsystem limitations.

The Importance of Memory in Terminal Server Implementations

Memory is a key enabler in any terminal server deployment. Strictly speaking, the amount of memory in a terminal server will directly impact the number of concurrent users that terminal server can support.

In a series of tests performed by Microsoft in support of the published whitepaper "*Terminal Server Scaling and Performance on x64-Based Versions of Windows Server 2003*", Microsoft compared two identically configured terminal server environments, one each on 32-bit and 64-bit versions of Windows, to determine user density capabilities. One of the key findings at the conclusion of the testing was that although 32-bit applications performed just as well on a 64-bit platform as they did on 32-bit operating systems, the memory requirements of running the same 32-bit applications on x64 Windows terminal servers almost doubled, largely due to the increased memory data structures associated with 64-bit architecture. As a result, Microsoft recommends increasing the amount of RAM used in 64-bit terminal servers by 1.5 to 2 times the capacity typically used in 32-bit implementations to accommodate the increased memory demands expected from running 32-bit applications on a 64-bit platform.

Factors that will affect the memory requirements for terminal server include the number of active concurrent users, the type of applications executed by each user and the number of concurrent applications each user will run within their individual sessions. More memory is typically required for newer, more-advanced versions of staple applications such as Office 2007 and Adobe Acrobat Reader 8 when compared to earlier versions of the same products. Furthermore, processor and operating system architecture will play a large part in determining user density, which will directly translate into the memory requirements of the terminal server.

Memory Provisioning Recommendations

Windows Server 2008 opens a whole new realm of possibilities in terms of user density per terminal server. With Standard Edition of Windows Server 2008 now available in a 64-bit architecture, organization will have the ability to realize the greatest ROI potential from a Microsoft terminal server by provisioning enough memory.

A typical Windows Server 2008 operating system running Terminal Services will require approximately 350-400 MB as a baseline. Other factors to consider in memory capacity planning include the number of users, the number of applications per user, the type of application and how the application will be used.

> **Number of Users** – Typically, if users will be running RemoteApp terminal server applications and will not be connecting to a desktop session, estimate approximately 10 MB per user as a

baseline. If users will connect to a published desktop to run their applications, increase this limit to 15 MB per user to handle the Windows Explorer shell-related binaries.

Applications – The individual memory requirements for each application must also be factored into the equation. For instance, Microsoft Excel consumes approximately 8MB of memory for each running instance's working set. However, complex client-server or line of business applications may require three to four times that amount, or possibly more. The requirements of each application to be hosted on a terminal server platform must be investigated and documented. Other factors include the type and size of documents used with the applications that may affect an application's working set in memory.

Number of Applications per User – Most likely, a terminal server environment will see differing levels of application use per user. Typically, these can be categorized into classifications such as "heavy users" that use many applications concurrently within a terminal server session, to "light users" that only run a single application at a time and connect intermittently.

Application Use Patterns – Another factor in memory requirements is the application use patterns. Many organizations have terminal server session activity 24/7, with users accessing the terminal server farm from different geographical regions and time zones. Additionally, application usage may peak at certain times, such as an increase in use of a financial application as the month's end approaches.

Each of these criteria must be weighed and calculated to determine the memory requirements of a terminal server running Windows Server 2008.

Memory is by far the most critical factor in determining user density on a terminal server. With 64-bit terminal servers becoming more mainstream, particularly considering the potential user density increases with 64-bit architecture, provisioned memory should be maximized. Where it was once common to find terminal servers with no more than 8GB RAM, most likely a 64-bit Windows Server 2008 terminal server will require twice or perhaps three times that amount to achieve user densities approaching several hundred concurrent users.

Server Core

Server Core is a new installation option for Windows Server 2008 x86 and x64 versions which provides a minimal running environment for specific server roles which significantly reduces the overall footprint of the operating system. The benefits of a Server Core installation include significantly lower disk space requirements, reduced memory overhead associated with traditional operating system graphical environment and a significantly reduced attack surface.

Windows Server Core installations support the following server *roles*:

- Active Directory Domain Services
- Active Directory Lightweight Directory Services (AD LDS)
- Dynamic Host Configuration Protocol (DHCP Services)
- Domain Name System (DNS) Server
- File Services
- Print Services
- Streaming Media Services
- Web Server (IIS 7.0)
- Hyper-V (Virtualization)

Server Core installations only install a subset of the traditional binaries required to support a particular server role. For instance, in a Server Core installation, the Windows Explorer shell is not installed. Instead, management is handled using the command line interface, or other traditional remote management techniques such as Remote Desktop access to the command line interface, Microsoft Management Console (MMC) or WMI scripting interfaces.

Benefits of a Server Core Installation

One of the major benefits of a Server Core installation is the significantly reduced attack surface. For example, since the Windows Explorer shell is not even installed on the system on a Server Core installation, security vulnerabilities exposed through the binaries associated with Windows Explorer are no longer a concern. This reduced operating system footprint has the added benefit of reduced maintenance associated with operating system security patches that would otherwise be required in a traditional Windows Server installation.

Furthermore, most infrastructure services listed above are not managed locally; rather typical management operations are performed using the MMC snap-in associated with the service, or are automated using scripts that manipulate the service through exposed WMI interfaces.

Server Core installations also have the added potential for increased performance over a similarly configured traditional Windows installation. This translates to an increased ROI as the CPU and memory resources in the hosting server hardware are directly assigned to the execution of the intended core services rather than to handling the overhead required to maintain the graphical Windows environment and other non-essential services. More of the memory installed in the host is available for the core services since non-essential binaries are not installed and therefore never loaded.

Finally, Server Core installations require far less disk space compared to traditional Windows Server 2008 installations. A typical Server Core base installation requires only 1GB of disk space, with an additional 2GB to handle role-specific operations, compared to between 6GB and 8GB for a traditional Windows Server 2008 installation.

Hyper-V

Hyper-V, formerly codenamed *Viridian*, is a new virtualization role included with Windows Server 2008. Hyper-V is technology that provides virtualization capabilities which enables multiple operating system instances to run concurrently on a single physical machine and is included with Windows Server 2008 Standard, Enterprise and Datacenter x64 Editions.

The first generation of Microsoft virtualization, Microsoft Virtual Server 2005 R2, was based on a hosted form of virtualization technology, where the 'host' operating system runs directly on the hardware, and the virtual machine monitor (VMM- the virtualization layer) accesses the hardware through the hosting operating system. Virtual machines are then provisioned and run on top of the VMM (see Figure 1, Hosted Virtualization Architecture).

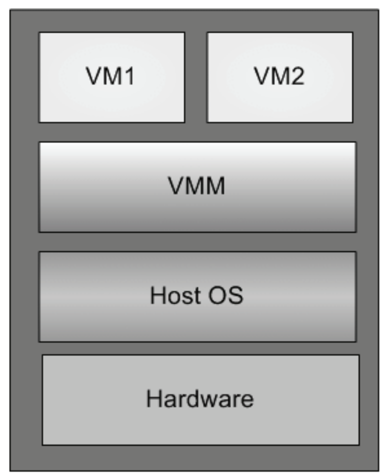

Figure 1, Hosted Virtualization Architecture

Hyper-V server virtualization technology takes advantage of, and requires, systems with hardware-assisted x64 platforms using either Intel VT or AMD-V processors that incorporate hardware data execution protection. Hyper-V architecture is based on a hypervisor, which is a layer of software that runs directly on the hardware and below the operating system. The function of the hypervisor is to enable the creation of isolated execution environments, known as partitions, and to arbitrate access to physical hardware resources for all running partitions. Each partition is assigned its own set of virtual resources, such as CPU, memory, disk, and network, and virtual machines (VMs), to a guest operating system.

A Windows Server 2008 Hyper-V server includes one instance of the parent, or 'root' partition; the name *root* comes from the fact that it's the first VM created, and all child partitions are then managed by the parent partition. The parent partition owns all resources that are not owned by the hypervisor, and is responsible for creating child partitions and managing their virtual hardware resources, exposing the Hyper-V management interface, power management, plug and play, managing hardware failures, and serving as the boot loader for the hypervisor (see Figure 2, Hyper-V Architecture).

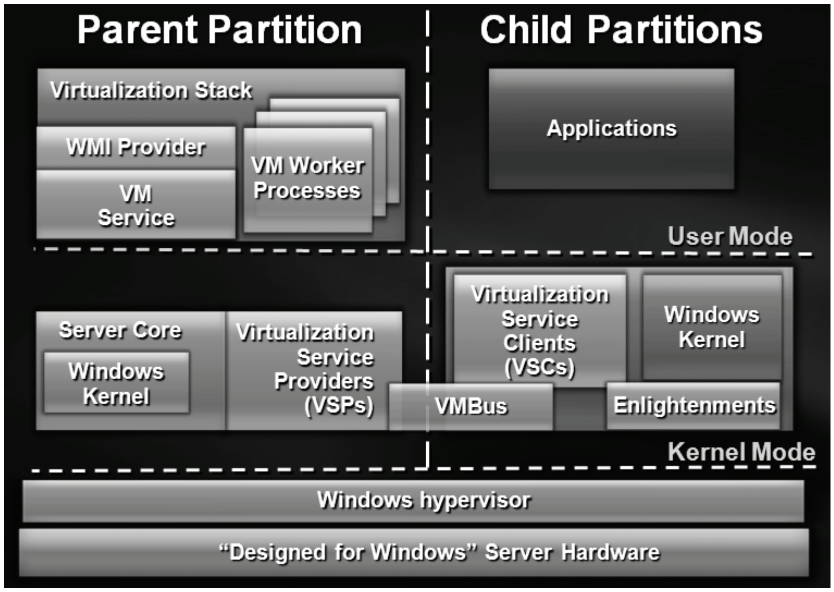

Figure 2, Hyper-V Architecture

From a memory usage perspective, it's important to consider the total memory requirements of the virtual machines hosted by the platform. Hyper-V supports the assignment of up to 64GB of RAM dedicated to each virtual machine. The maximum amount of physical RAM a Hyper-V server supports is dependent upon the hosting Windows Server 2008 edition and architecture. Table 4 details each processor architecture and Windows Server edition with the maximum memory addressable by Hyper-V.

	Windows Server 2008 Edition	Maximum Hyper-V Addressable Memory
X86	Standard Edition	4 GB
	Enterprise Edition	64 GB
	Datacenter Edition	64 GB
X64	Standard Edition	32 GB
	Enterprise Edition	2 TB
	Datacenter Edition	2 TB

Table 4 – Maximum Hyper-V Addressable Memory

No administrative limit as to the number of virtual machines running on Hyper-V exists; the limitations are based upon available hardware resources. It should be noted that no memory is shared between virtual machines, so unused memory resources assigned to virtual machines will not be accessible to other processes or virtual machines.

Memory Recommendations with Hyper-V

Virtual infrastructure places great demands on hosting servers for CPU and memory resources to maintain the virtualized workloads executing on the platform; therefore, memory is a key factor in the consolidation of x86 workloads in a virtual environment.

When planning for memory requirements of a Hyper-V server, it is important to consider the following:

Base system memory requirements for the Hypervisor and parent partition – These values are significantly lower for a Windows Server Core build as opposed to a traditional Window Server 2008 installation. A Server Core installation will enable more effective use of CPU and memory resources in the virtualization of x86 workloads rather than to hosting operating system tasks.

Memory requirements for the identified virtual machine workloads – Each virtual machine that will be hosted by the platform will require a dedicated amount of system memory.

Additional memory headroom for future virtual machine workloads – Consider the future workload of additional virtual machine, as well as changing conditions in existing virtual machines that may increase resource requirements.

Provided Windows Hyper-V host servers have enough free CPU, RAM, networking and storage resources to support additional virtual machine workloads, the increase in virtual machine-to-host density will translate to a reduction in datacenter TCO.

Properly sizing memory prior to deployment is critical, as having to add memory later to production Hyper-V servers may introduce unwanted down-time. Furthermore, improper initial memory sizing may lead to having to dispose of lower capacity memory modules as higher-capacity modules are added in their place. Consider the existing needs of platform as well as future workload demands, and whenever possible, provision virtual servers with as much RAM as economically possible. The costs associated with adding additional RAM to Hyper-V host servers to facilitate an increased virtual machine density should not be a significant consideration; generally speaking, adding new physical servers to the infrastructure will always increase the datacenter's TCO significantly more than adding more memory to existing Windows Hyper-V servers to support additional virtualized workloads.

Finally, sizing memory workload demands might require a '*provision-then-monitor*' approach, unless there is reliable historical performance data available. Therefore, as a best practice, consider following existing memory provisioning guidelines for Hyper-V virtual machines and monitor actual memory usage to determine if more or less assigned memory is warranted.

Microsoft Visual Studio 2008

Microsoft Visual Studio 2008 is the latest version of the Microsoft Integrated Development Environment (IDE). Visual Studio 2008 can be used to develop console and GUI-based applications for all platforms supported by Microsoft, including Windows, Windows Mobile, .NET Framework, .NET Compact Framework, and Microsoft Silverlight, as well as directly into certain applications, such as SQL Server and Microsoft Office. Visual Studio also includes a code editor supporting a technology known as

IntelliSense (code auto-completion), code refactoring, and source and machine level debugging, for built-in languages C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), and C# (via Visual C#). For all of this integrated and highly valuable functionality, Microsoft recommends a minimum of 1GB of RAM for the system running Visual Studio 2008, and anecdotal evidence suggests that for a smoothly responsive system, such as real time IntelliSense response, 2GB is recommended for the system. However, as described below, there are strategies that can significantly decrease the development timelines and work effort and having ample memory available is the key to success.

A use-case that is gaining in popularity for development environments, particularly off-shore development talent, is exposing the IDE and protecting the source code via Virtual Desktop Infrastructure (VDI). The core design element of VDI is multiple desktop instances running on a Windows Server Virtualization platform or hypervisor within a virtual machine, while users access the session over the remote desktop protocol (RDP), such that all applications, processing, and data are securely confined to the datacenter and are never actually present at the user's workstation. The value of this model increases with the addition of server platforms or applications made directly available to the developer for development and testing purposes; as opposed to provisioning a physical SQL Server that several developers must share for the purpose of developing or testing code, a virtual machine can be provisioned with SQL server pre-installed in minutes and then suspended when not required, a task that is far more difficult with a physical server.

The approach of using VDI to host Visual Studio 2008 requires the combination of a server virtualization platform and a great deal of advanced planning. Therefore, determining the memory requirements is an extension of the process in determining the memory needs for Hyper-V (described above). Simply determine the memory requirements for the desktop operating system, add the additional memory requirements for Microsoft Visual Studio 2008 and multiply the total by the number of concurrent VDI user sessions.

The only drawback in the approach of leveraging a VDI environment is the requirement of directly-connected network access, since the user cannot access their session without interactive connectivity to their desktop session via RDP.

The Portable VDI Environment

In the case where VDI is not feasible, such as when it may not be possible for developers to remain connected to the VDI infrastructure during development or debugging, a compelling option is to create the 'portable VDI' environment.

Microsoft Virtual PC, available as a free download, enables the user or developer to create multiple virtual machines that run directly on their laptop or desktop PC. A fully functional environment would consist of the host operating system on the laptop running Virtual PC, a virtual machine running an installation of Windows Vista and Visual Studio 2008, a third virtual machine running Server 2008 and SQL Server 2008, and perhaps IIS 7.0. Naturally, the actual number and purpose of each virtual machine would vary based on the needs of the application, but the concept remains the same regardless of the circumstances.

This configuration could be supported using virtual machines with anemic memory footprints on a laptop with 2GB of RAM with minor-to-moderate performance degradation, assuming the processing of the application itself was manageable. However, a more optimal approach would include a host laptop or desktop provisioned with 4GB or more of memory which could easily support these same virtual machines with a much more memory assigned to each, reducing any performance drag of the development environment.

SQL Server 2008 and Data Services Consolidation

Organizations are deploying an ever increasing number of applications to provide new services, manage business processes, and to gain a competitive advantage through analysis and data mining. The number of application servers and data storage systems required to support this ever increasing number of applications has grown dramatically since Microsoft released its first home-grown version of SQL Server in 1999. The hardware and operational costs of supporting this growing burden of systems infrastructure has grown into a significant financial and operational burden.

In order to address these many operational and financial challenges associated with the backend database infrastructure, the SQL Server team improved upon several capabilities that aid in SQL Server consolidation.   Through SQL Server consolidation, the IT organization can reduce the number of physical servers in the network, thereby saving on operational costs, avoiding unnecessary data center expansion, lowering the data center cooling and power requirements, and improve upon operational efficiency, as measured in terms of managing the same or growing amount of data center processing demands with the same or fewer IT head-count. Proper capacity planning, particularly around memory requirements, is the key to a successful consolidation initiative.

Microsoft SQL Server 2008 consolidation capabilities fall into the following three scenarios:

- Consolidating SQL Server Instances with Multiple Databases
- Consolidating Physical Servers with Multiple Instances
- Consolidating Data Services Through Virtualization

Each of these scenarios is further explained below.

Consolidating SQL Server Instances with Multiple Databases

The most direct approach to consolidating data services with SQL Server 2008 is to use a single instance of SQL Server with multiple databases. This approach is suitable when all of the databases within the consolidation scope have similar security, manageability, and compatibility requirements, and the hardware can provide the required level of performance and scalability for the workloads that are generated in all of the databases.

The numbers of instances supported by each edition of SQL Server 2008 are shown in Table 5 below:

Edition	Maximum Instances
SQL Server 2008 Standard Edition	16
SQL Server 2008 Enterprise Edition	50
SQL Server 2008 Developer Edition	50

Table 5

Consolidating Physical Servers with Multiple Instances

When consolidating databases with different security, manageability, or compatibility requirements, consolidation of these data services is achieved by running multiple instances of SQL Server 2008 concurrently on a single physical computer to reduce the hardware costs, licensing costs, and administrative overhead. The instances are completely isolated from each other and changes to one instance do not affect other instances on the same computer. As well as reduced hardware costs through consolidation, another benefit is reduced licensing costs because only one SQL Server license per physical processor is required regardless of how many instances are installed.

Consolidating Data Services through Server Virtualization

For complete isolation at the operating system level, SQL Server 2008 supports server virtualization. Leveraging Microsoft Virtual Server or Hyper-V, multiple virtual machines with completely separate operating systems can be installed on one physical machine. When this approach is combined with Microsoft Windows Server 2003 R2 Datacenter Edition and SQL Server 2008 Enterprise Edition, only one Windows license and one SQL Server license is required per physical processor, regardless of how many virtual machines are installed on the physical server.

The hard disks of each virtual server exist as files on the host operating system, which contributes to simplified backup, relocation or deployment, and provides an ideal environment for development and testing. Furthermore, many organizations maintain strict policies around combining application environments such as development/test and production. SQL consolidation through virtualization is the only method that can satisfy the financial and business-related benefits of SQL consolidation and still comply with corporate policy around segregation of application environments.

The greatest level of isolation between database solutions with different workloads, security requirements, manageability needs, or compatibility requirements is possible by consolidating data services with virtualization, while minimizing the number of servers and licenses required and simplifying network infrastructure.

Memory Considerations for SQL Server Consolidation

The memory specific considerations of SQL Server consolidation are focused on providing enough working memory to execute SQL queries of all the consolidated SQL workloads without sacrificing performance. This approach will likely require a target consolidation server with significantly more memory than on any of the original SQL Servers, though there might be opportunities to redeploy memory from the SQL Server decommissioned through consolidation.

While optimizing the performance of consolidated data services can present many challenges, the easiest approach to ensuring success it to provide plenty of hardware resources to the consolidation platform, including multiple high-performance processors and large amounts of memory. In addition, several options exist in SQL Server 2008 for controlling and managing performance.

Resource utilization can be tightly controlled through the SQL Server 2008 Resource Governor. This mechanism enables organizations to throttle the CPU and memory resource utilization of different workloads to prevent a single instance or operation from consuming too many resources and affecting overall system performance.

SQL Server 2008, coupled with 64-bit editions of Windows Server 2008, can support more hardware resources then previously possible, with resource limitations restricted only by the host operating system capabilities. Additionally, SQL Server 2008's ability to support dynamically adding both memory and CPU resources without the traditionally associated downtime provides a much more flexible and robust platform for data consolidation scenarios.

The use of server virtualization provides an efficient consolidation strategy, because a single robust x64-based server platform can provide a single footprint solution to a plethora of solution requirements, from 32-bit SQL Servers with 4GB of RAM through 64-bit SQL Servers with 32GB of RAM. Virtual machines can be deployed faster than physical servers, and the management of many workloads on few physical servers for some customers is simplified by providing one interface and set of operational processes, as opposed to many distributed systems with diverse management, provisioning, and operational considerations.

Summary

Microsoft Windows Server 2008 marks a new evolution in the Windows Server system, enabling server virtualization, more accessible 64-bit platforms, increased server security and better manageability to enterprise environments. Coupled with increased memory capacities and advances in memory management, Windows Server 2008 will enable organizations to reach new levels in server functionality and scalability while offering a more compelling return on investment.

64-bit versions of Windows Server 2008 are now offered in Standard edition with much higher memory capabilities. Microsoft has recognized the inherent performance benefits of 64-bit architecture and has made strategic endeavors to ensure that enterprise server applications realize the greatest performance gains from leveraging 64-bit technologies; this is apparent in Microsoft's bold move to offer the latest generation of their enterprise messaging platform, Exchange Server 2007, only in 64-bit implementations for production environments. Furthermore, Hyper-V builds on 64-bit architecture to enable affordable high-performance virtual infrastructure for the consolidation and containment of x86/x64 workloads.

At the heart of all of these benefits, memory is a key enabler. Provisioning sufficient memory in server hardware will increase both the efficiency and scalability of Windows Server 2008 platforms and enterprise applications. To decrease the TCO of high memory capacity implementations, consider Kingston memory rather than OEM-procured RAM.

With high performance multi-core processors available today, memory is often becoming the limitation in many applications, including virtual machine density and Microsoft-based services. By provisioning high memory headroom into physical servers prior to deployment, IT organizations can optimize their Windows environment and further reduce their data center's TCO while achieving greater ROI on each server deployed.

About Kingston® Memory

When you need reliable server memory, count on Kingston®. Kingston modules are engineered from premium components and backed by guaranteed compatibility, free 24/7 tech support and a lifetime warranty. You can rest assured that they've passed strict testing, including our patented burn-in testing process. For added peace of mind, our modules work with all of today's server diagnostic pre-failure warning software. KingstonCare, our free suite of services, completes the package with on-site spares, cross-ship RMAs, service cost reimbursement and an easy-to-use website that takes the work out of submitting service requests.

About the Author

Michael Burke is the Microsoft Practice Director for a leading Virtual Infrastructure Services company headquartered in the Northeast, providing virtualization design and integration services nationwide. Burke has over 10 years experience working closely with Microsoft products. He has written many technical articles and whitepapers around Microsoft products and technologies and has been a guest speaker at several international conferences on the subject.

Printed in the USA MKMS – 1068