Consolidating Microsoft SQL Server 2008 on HP Integrity servers

Executive summary

With the release of Microsoft® SQL Server 2008 and the end of mainstream support for SQL Server 2000, many organizations are moving to a 64-bit database platform and consolidating their 32-bit SQL Server 2000 databases to reduce the number of physical servers deployed.

HP Integrity servers running Microsoft Windows® Server 2008 and SQL Server 2008 provide a scalable database platform that is ideally suited for consolidation, delivering the flexibility, scalability, and availability needed in this environment, while also offering a broad range of consolidation options.

SQL Server 2008 provides a key feature – Resource Governor – that removes a barrier to consolidation by allowing limits to be imposed on resource usage, preventing one application from impeding the performance of another. This is very useful in a consolidated environment where workloads can grow at different rates, some very quickly and others at a more moderate growth pace.

While some applications may not be ready for SQL Server 2008, they can still benefit from consolidation to reduce the consumption of power, cooling, and human resources. Although earlier SQL Server versions do not offer the Resource Governor feature, this document also explores alternative methods for effective workload management in a consolidated environment.

While each organization is unique, with its own challenges, HP Integrity servers can provide solid platforms for today's most popular versions of SQL Server – SQL Server 2000, SQL Server 2005, and SQL Server 2008. The reliability, scalability, and flexibility delivered by HP Integrity servers allow customers to undertake SQL Server consolidation projects with confidence.

This document describes how to consolidate SQL Server instances, with a particular focus on SQL Server 2008 and the HP Integrity line of servers. Performance testing of a simulated consolidation is described.

Intended audience: This document is intended primarily for database administrators and other IT professionals interested in consolidating a SQL Server environment.

This white paper describes testing performed in September, 2008.

Introduction

SQL Server has proven to be an easy-to-use, reliable, scalable data management platform that has, over the years, been employed by many independent software vendors (ISVs) as a back-end database, driving up demand for SQL Server across all customer segments. However, this wide usage has often led to server sprawl as each ISV's application tends to call for its own SQL Server database server; alternatively, the organization may have been growing so fast that it seemed easier to purchase additional servers whenever a new project came along. Whatever the reason, SQL Server sprawl is the end result, leading to significant under-utilization of server resources.

In a recent industry analyst study, respondents rated the top reasons for consolidation as follows:

1. Better management of IT assets – 41%
2. To achieve better total cost of ownership (TCO) – 28%
3. To achieve better service agility – 17%

Thus, this study indicates that organizations want better control over their information assets – a goal that can be achieved by consolidation.

TCO

Reduced TCO may often be the most compelling reason for an organization to adopt consolidation, an approach that can often lead to immediate cost savings. For example, by eliminating processors and servers, consolidation can reduce software and maintenance costs; it can also create savings in the data center by freeing up rack space, cutting down cooling and electricity costs, and – for some organizations – reducing leasing costs. Consolidation can also reduce the workload for over-burdened IT staff by making the environment easier to manage.

Server requirements

When the number of servers in an environment is reduced, the need for reliability and stability increases, since fewer servers are now supporting more business functionality and more users. Combining workloads in a consolidation environment can lead to variable growth, which may be difficult to support – especially when new applications with no history of workload behavior or growth are added. As a result, the platform supporting the consolidation environment must be scalable, with predictable performance. The HP Integrity line of servers is ideally suited to meet these requirements, having been engineered from the processor level up to be a fault-tolerant, reliable, scalable server platform.

Current conditions

Although SQL Server 2008 was released in August 2008, SQL Server 2000 continues to be used in an estimated 40 – 60% of SQL Server production environments, even though mainstream support has been withdrawn. Although SQL Server 2005 was introduced in October 2005, its widespread adoption has been limited by the reluctance of ISVs to upgrade their database platforms, tying customers to SQL Server 2000 until the ISV decided to upgrade.

As databases for ISV and home-grown applications have developed over the years, the 32-bit SQL Server 2000 platform has been solid. However, in order to maintain service level agreements (SLAs) and provide high levels of performance, database administrators (DBAs) and developers have tended to dedicate a physical server to each application and database. This approach typically results in SQL Server sprawl, whereby many servers are serving single – or a very few – databases.

In addition to server sprawl, security is another issue in today's computing environments, often making the isolation a key concern. For example, one application and database may be strongly protected while others may be less secure, leading to an increasing demand for isolation, not only at SQL Server instance level but also at the server- and storage-levels.

In response, security models for SQL Server 2005 and 2008 have been made significantly stronger, reducing the risk of a compromised database impacting other databases and their associated applications. In turn, stronger security has reduced the need to separate and isolate databases. For environments where enhanced SQL Server 2005 and 2008 security features are not enough and isolation is required, cell-based HP Integrity servers can be carved up to create separate servers that are electrically isolated.

Memory access

Since the vast majority of SQL Server 2000 installations are 32-bit, the amount of memory that SQL Server can address natively is often limited to 4 GB of virtual address space. By default, 2 GB is allocated to user-mode processes, such as SQL Server, and 2 GB is allocated to support kernel-mode access by the operating system.

By using the /3GB switch at boot time, it is possible to reallocate the 4 GB of virtual address space to provide 3 GB for user-mode processes and limit kernel-mode access to 1 GB.

32-bit SQL Server platforms support Address Windowing Extensions (AWE) through the use of Physical Address Extension (PAE) to provide 36-bit addressing (up to 64 GB of physical memory). However, accessible memory is mapped to the original virtual address space and can only store data pages within it. Thus, for example, AWE memory cannot alleviate pressure on procedure cache memory, sort memory, or connection memory.

With applications becoming more and more demanding, memory addressing constraints make it unrealistic to consolidate SQL Server instances on 32-bit platforms. However, 64-bit platforms provide up to 16 TB of virtual address space, significantly more than the 4 GB available by default on 32-bit platforms. Thus, memory access on today's 64-bit servers is fast enough to alleviate the pressure on all components of SQL Server memory, including procedure cache and connection memory.

Memory access is the key to database processing performance. Database data is stored on disk and can be accessed in milliseconds; however data that is being used by the application and may be accessed again is stored in cache memory, which can be accessed in nanoseconds. Since there are one million nanoseconds in a millisecond, the speed difference between disk I/O and memory access is significant[1].

Note:
Even on desktop and personal systems, memory is often the key to improving performance. You should avoid disk I/O whenever possible.

To summarize, the use of a 64-bit platform is typically an integral part of the solution when addressing SQL Server sprawl. This platform allows SQL Server to take advantage of available memory and minimize the need for disk I/O. Up to 2 TB of RAM can be addressed on the HP Integrity server line, enabling more efficient data access over the course of business operations.

SQL Server versions

This section outlines the capabilities of SQL Server 2000, 2005, and 2008.

SQL Server 2000

As stated earlier, SQL Server 2000 is the SQL Server platform that is most commonly used today. It was first released in the year 2000 as a 32-bit database platform, typically in a Standard or Enterprise Edition. While these editions offer different features and functionality, the basic engine is the same.

Note:
Other, rarely-used editions of SQL Server 2000 have more restrictions than the Standard Edition.

The Standard Edition of SQL Server 2000 limits the engine to 2 GB of RAM and four processors. Even with the advent of multi-core processors, this limitation restricts the scalability of this edition.

The Enterprise Edition removes some of the barriers to scalability. The 32-bit version of Enterprise Edition supports up to 32 processors and 64 GB of RAM, while the 64-bit version supports up to 64 processors and up to 512 GB of RAM.

[1] To put this in perspective, one million seconds equates to approximately 11.5 days.

Standard and Enterprise Editions of SQL Server 2000 are limited to 16 instances[2] of SQL Server, where each instance may server multiple databases. Support for multiple instances each serving multiple databases can provide an isolation mechanism when required by security concerns or regulations.

There are no additional licensing costs associated with installing multiple instances of SQL Server on a single server since licensing for SQL Server 2000 is either server-based – with client access licenses (CALs) – or processor-based – no CALs. Both licensing modes allow the bits to be installed multiple times on the same server.

The 64-bit version of SQL Server 2000 is only available for Intel® Itanium® processor-based server platforms, such as the HP Integrity server family. While there is no x64 version of SQL Server 2000, the 32-bit version may be run on an x64 platform, as long as SQL Server is running in Windows 32-bit on Windows 64-bit (WoW64) mode. With WoW64, SQL Server can take advantage of 4 GB of virtual address space; however, it must use AWE memory to see beyond 4 GB. Thus, the consolidation of SQL Server 2000 on an x64 platform, while plausible, would be limited due to the constraints of the native 32-bit code.

All HP Integrity servers can run SQL Server 2000 natively without WoW64 and can access up to 512 GB of RAM, a factor that, alone, increases the potential compression ratio for databases and instances.

SQL Server 2005

SQL Server 2005 was released in November 2005. Standard and Enterprise Editions are provided in the following flavors:

- 32-bit
- x64
- Itanium

Enhancements

SQL Server 2005 has key enhancements over SQL Server 2000 in the following areas:

- Database engine
- Memory managements
- Processor load balancing
- I/O control

In addition, SQL Server 2005 is Non-Uniform Memory Access (NUMA)-aware, providing enhanced load management and memory access capabilities for cell-based system such as the mid- to upper-range of the HP Integrity server line.

SQL Server 2005 also introduced the Hot Add Memory feature, which allows memory may be added to a system without a reboot, thus reducing downtime.

[2] An instance is an installation of the bits.

Restrictions

As with SQL Server 2000, Standard and Enterprise Editions of SQL Server 2005 have different restrictions. Standard Edition limits the number of processors to four, while Enterprise Edition can support as many processors as the operating system is able to address.

Note:
While a processor may have multiple cores, it is still considered to be a single processor for licensing purposes.

Windows Server 2003 and initial releases of Windows Server 2008 have a limit of 64 logical processors, which maps to 64 single-core processors or 32 dual-core processors. If hyper-threading is utilized, the maximum number of processors supported is reduced by half.

Note:
The current Windows Server 2008 release (October 2008) limits the number of threads that may be scheduled to 64.

SQL Server 2005 removes memory limitations in both Standard and Enterprise Editions.

Standard Edition is limited to 16 instances and can support a two-node cluster, while Enterprise Edition can support up to 25 instances in a cluster of up to eight nodes or up to 50 instances in a non-clustered configuration.

Enhanced resource utilization

SQL Server 2005 supports dynamic memory reconfiguration and CPU affinity, which can enhance resource management. For example, consider a set of databases that are busiest in the morning hours between 6am – 11am; another set of databases is busier in the afternoon hours between 12:30pm – 4pm. You could encapsulate each set of databases in a SQL Server instance, with the first instance being allocated more resources during the busy morning hours. A job can be automatically scheduled to de-allocate memory and processor resources, then shift them to the second instance during the afternoon hours to better utilize resources.

Logical processor grouping

SQL Server 2005 can take advantage of Soft-NUMA technology, which is essentially a logical grouping of processors that may be affinitized based on a particular server connection port. While this technology can enable creative workload management, it does not effectively limit one workload as opposed to another when both server connections have processors in common.

The Resource Governor feature introduced in the Enterprise Edition of SQL Server 2008 provides a much more effective mechanism for managing workloads. However, affinitization can be used to enhance resource utilization in environments where SQL Server 2008 has not been adopted.

SQL Server 2008

SQL Server 2008 was released in August 2008, effectively completing and refining the third-generation SQL Server product, while adding valuable features that facilitate consolidation.

Note:
SQL Server 6.0 and 6.5 are considered to be first-generation products; SQL Server 7.0 and SQL Server 2000 are second-generation; and SQL Server 2005 and SQL Server 2008 are third-generation.

Enhancements

While there are many new features and enhancements in SQL Server 2008, the following have the most impact on consolidation:

- Resource Governor – Removes a barrier to consolidation by allowing limits to be imposed on resource usage, preventing one application from impeding the performance of another; for more information, refer to the Resource Governor section
- Database encryption – Strengthens security
- Database compression – Enhances storage management
- Policy-based management – Allows standards and best practices to be enforced
- PowerShell – Allows a single command to be programmatically scripted across multiple databases, instances, and servers.
- Hot Add – Provides the capability to hot add processors as well as memory

Editions

While SQL Server 2008 editions are similar in nature to those of SQL Server 2005, the following differences should be noted:

- Enterprise Edition increases the number of supported cluster nodes to the maximum number of nodes supported by the particular Windows operating system being used; Standard Edition maintains its two-node clustering ability
- Backup and data compression capabilities are only available with Enterprise Edition
- Standard Edition only provides limited support for database encryption

To summarize, the features and functionality provided by SQL Server 2008 Enterprise Edition makes this the best platform for consolidation. If you are not ready to migrate your environment to SQL Server 2008, SQL Server 2000 and SQL Server 2005 still provide excellent database platforms and can support consolidation in a variety of IT environments.

The next section describes how to collect data before embarking on a consolidation project.

Data collection

When tackling a consolidation project, you should learn as much as possible about the target environment so that you are able to design an architecture that can accommodate the demands of your various workloads, and the future growth of these workloads. This requires you to collect data in areas such as server performance, demographics and configuration parameters for servers and instances, and business rules.

Performance data

Performance data should be collected on every server and SQL Server instance using tools such as Windows Performance Monitor (Perfmon), which allows you to characterize the performance of individual components within a particular system. The end result of performance data collection is a performance profile for the servers and instances observed, allowing you to rate each server in the appropriate performance band, Tier 1, 2, or 3.

These performance bands are specified purely from a performance perspective and not business criticality. The three tiers are described as follows:

- Tier 3 – Tier 3 servers are characterized by light loads and excellent performance. In many environments, Tier 3 servers dominate the landscape, making up 60% – 75% of the server population. Servers in such environments tend to be under-utilized, making them excellent candidates for consolidation.
- Tier 2 – Tier 2 servers have moderate loads, moderate performance, and moderate consolidation potential.
- Tier 1 – Tier 1 servers are heavily loaded, running near their performance thresholds; you should only consolidate such servers after performance issues have been resolved. Tier 1 servers usually make up the smallest population, with 5% – 15%.

Note:
Many organizations impose a single standard for servers added to their environments. As a result, database servers are often overpowered with processor resources but undersized in memory and storage.

You should collect performance data over, at minimum, a 24-hour period during a typical workday or, perhaps, a period of heightened activity such as month-end processing. It is very important to collect data with a typical workload since processing capacity and memory requirements are based on current observed values as well as growth information provided by business rules. If you were to collect data collection while the workload was unusually light, the resulting solution would have little or no headroom to support growth.

Remember that the busiest time of day for processor workload may not coincide with the busiest time for other key components such as memory and I/O. Moreover, do not restrict your focus to peak values: if, for example, processor utilization spikes to 95% for a period of 10 seconds but fails to exceed 20% for the rest of the day, a solution based on 95% utilization would lead to significant resource wastage.

One way to determine a reasonable utilization rate involves dividing up the observation period into meaningful segments. If, for example, you are monitoring a 24-hour period, you can cut this period into 24 one-hour segments and obtain an average utilization rate for each. Monitoring MAX AVERAGE Perfmon counter values for these periods can provide a reasonable indication of the resources that would be needed in a consolidation environment.

Of course, the 24-hour observation period could just as easily be sliced into, say, 48 30-minute segments or six four-hour segments. Note, however, that with shorter segments, a spike in utilization has more impact on MAX AVERAGE. For example, although the average CPU utilization over a 24-hour period may be 8%, slicing this period into 24 one-hour segments may generate a MAX AVERAGE reading of 15%. With shorter segments, MAX AVERAGE might become 80% or more. Thus, if you slice the observation period too finely, your consolidation environment may be over-sized; if you slice too broadly, your environment may be underpowered.

Some data collection efforts require multiple servers to be monitored over a longer period of time – up to a month in some cases – to identify peaks of activity and periods of sustained, higher than normal activity.

Note:
Long-term data collection efforts often involve the installation of a monitoring server or the deployment of agent software on targeted servers, a process that can take weeks depending on standard operating procedures or security concerns in the environment being analyzed.

For example, payroll processing systems are often busiest on Mondays and Tuesdays, the days typically used for data entry, verification, and check processing. If the short-term data collection described earlier were not performed on Monday or Tuesday, the consolidation environment would be under-sized. Thus, selecting appropriate time periods for data collection requires some basic intelligence – often derived from the database administration team – on periods that are busier than others. If this basic intelligence does not exist, a longer monitoring period would be beneficial.

In addition to the time period selected, sampling rate is also important in data collection. A very frequent sampling rate (such as one second) would generate vast amounts of information and might even impact the performance of the system being monitored. However, a less frequent sampling rate (such as 30 seconds or one minute) might miss sustained bursts of activity. A sampling rate of five or 10 seconds is sufficiently frequent to discover performance issues yet infrequent enough to keep data file size manageable and minimize the impact of data collection on the server.

Processor performance monitoring

The following Perfmon performance counters are typically monitored to characterize processor performance:

- **\Processor\% Processor Time**
- **\System\Processor Queue Length**
- **\System\Context Switches/sec**

While performance troubleshooting is outside the scope of this document, you should note that it can sometimes take the combination of several key counters to point to pressure points within the observed system.

Note also that one processor in a system may be more heavily loaded than others. At times, a particular processor may be allocated an operating system task such as managing network throughput, which would drive the utilization level of this processor higher. In systems with fewer processors, this imbalance may be more pronounced.

When monitoring CPU pressure, **% Processor Time** is a key counter. If **% Processor Time** approaches or exceeds 80% for a sustained period of time, you should be concerned.

Memory performance monitoring

More Perfmon counters are typically used to characterize memory performance than processor performance. Memory-related counters include the following:

- **\Memory\Pages/sec**
- **\SQLServer:Buffer Manager\PageLifeExpectancy**
- **\SQLServer:Buffer Manager\LazyWrites/Sec**
- **\SQLServer:Cache Manager**
- **\SQLServer:Memory Manager**

Using these – and other – counters, your main objective is to detect memory pressure in each area of memory. Again, overall results are not derived from one particular counter; rather, a combination of counters is used to present an overall view.

For example, if **PageLifeExpectancy** counter approaches or goes below 300, there is server memory pressure on the system. If you detect memory pressure, then your consolidation environment may benefit from a move to a 64-bit platform, which can address more memory and thus enhance performance.

I/O and network performance monitoring

I/O and network-related performance counters can provide another dimension to the performance profile. The following counters are often used:

- **\PhysicalDisk(*)\Disk Transfers/sec**
- **\PhysicalDisk(*)\Disk Read Bytes/sec**
- **\PhysicalDisk(*)\Disk Write Bytes/sec**
- **\Process(sqlservr)\IO Data Operations/sec**
- **\SQLServer:Databases(*)\Transactions/sec**
- **\Network Interface(*)\Bytes Total/sec**

Since it is often the slowest area of a server, disk I/O may create a bottleneck on many systems. Thus, you may need a specialist such as a storage array network (SAN) administrator to analyze Perfmon counter values for throughput and I/Os per second (IOPS) and then create a suitable storage solution for the consolidation environment.

There are many potential bottlenecks between SQL Server and physical spindles on the SAN. For example, if data requested by the server is residing on a storage medium, the speed of that medium has a significant impact on performance. While I/O channels can limit performance in some environments, HP Integrity servers are designed to eliminate as many I/O bottlenecks as possible; for example, each processor has a dedicated PCI bus slot available to provide a direct conduit to the storage subsystem.

While network speeds have improved dramatically over the years, the network can be a bottleneck. Thus, it is crucial for servers in a consolidation environment to be equipped with an appropriate number of suitably configured network cards.

In an environment with a SAN, the **Avg Disk Queue Length** counter has little meaning; instead, you should focus on **Avg Disk sec/Transfer**. If this value is above 20 milliseconds, there is cause for concern.

Demographic data

Each SQL Server instance has configuration settings and other demographic data that should be captured for analysis. In most cases – particularly for 32-bit servers with memory resource limitations that often make it impractical to run multiple instances – SQL Server systems run a single instance, usually an unnamed default instance.

The retrieval of demographic data from a SQL Server instance is usually accomplished using the standard SQLDIAG utility, which can retrieve a large amount of information that is valuable for consolidation analysis. While the type of data captured by SQLDIAG is configurable, default data includes the following:

- Recent error logs
- **Sp_configure** information
- Database information (such as names, size, and collation)
- Blocking information

SQLDIAG can be used to show how each SQL Server instance is configured, allowing you to identify differences between configurations. Some of these differences (such as Max Server Memory settings – especially when a single instance is involved) may be unimportant when consolidating an SQL Server environment. However, differences in Maximum Degree of Parallelism (MAXDOP), Cost Threshold for Parallelism, and collation may present a more significant challenge because these settings are configured at the instance- rather than database-level. Furthermore, MAXDOP affects the number of processors used and may thus impact operations on other databases.

The SQLDIAG output also includes error logs that can reveal current production problems that may not be revealed by performance counters: for example, when a filegroup or log file becomes full, an error code appears in the error log.

Business rules

The third objective for data collection is to discover the business rules associated with applications, databases, and the overall environment. If, say, an organization is running 100 SQL Server instances, it is quite possible for there to be as many as 100 sets of rules associated with them. However, when you are implementing a consolidation environment, you should develop and enforce standards to make management and support more efficient.

Taking into consideration the cost of uptime, you should specify the business criticality of each application using categories based on acceptable levels of service and downtime. In this way, you help define high-availability requirements for the consolidation environment.

You must also clearly define your tolerance for data loss, thus helping specify the disaster recovery capabilities required in the consolidation environment.

In addition, you will discover if an application layer is needed to reside and run on the database server. In rare cases where this layer is needed, you should ensure the application layer can be compiled for 64-bit operation to optimize server performance.

Other areas

Other areas for data collection include security, storage, and the compatibility of database platforms with ISV packages.

Security

Security is a major concern when creating a consolidation environment – not so much application layer security as the security surrounding the access and management of data. In some industries, there are rules and regulations that require separate management for different types of data; thus, for example, DBAs may be permitted to manage data for one part of the organization but not another. The only way to discover this type of information is to interview appropriate personnel across the organization.

Storage

Additional information on storage may be required as part of a consolidation initiative, especially the current environment includes a SAN.

As data is collected via Perfmon, you may need some additional information to place values obtained from storage-related counters in the proper context. For example, knowing the number of spindles behind a particular logical unit number (LUN) or drive letter can give more meaning to the **Avg Queue Length** for a particular drive. Occasionally, this exercise can reveal storage configuration concerns, such as data and log files residing on the same LUN.

Compatibility

Often, a consolidation initiative involves upgrading the database platform to the latest version, which may or may not be supported by a particular ISV package. Thus, compatibility data may impact the architecture of the consolidation environment.

You should categorize each application as follows:

- Fully supports SQL Server 2008 – No change is need on the client side; runs in native database compatibility mode ('10' for SQL Server 2008)
- Fully supports SQL Server 2008 – No change is needed on client side; must run in an earlier database compatibility mode ('9' for SQL Server 2005, '8' for SQL Server 2000)
- Fully supports SQL Server 2008 – Requires a change to the client application; may need to run in an earlier database compatibility mode
- Does not support SQL Server 2008 – Retain SQL Server 2005 SP2 instance
- Does not support SQL Server 2008 – Retain SQL Server 2000 SP4 instance

Furthermore, for each application that falls into one of the final two categories, you must decide whether to retain this application or search for another application that does support your database platform. You should assess the risk that support for these legacy applications may be withdrawn in the future.

Guiding principles for consolidation

This section outlines guiding principles issued by Microsoft for consolidating SQL Server.

Only consolidate homogenous workloads on a single Windows instance

The advice to only consolidate homogenous workloads on a single Windows instance is not limited to keeping SQL Server separate from other consolidation efforts, such as those involving file/print services or Microsoft Exchange Server. You should also separate the types of workloads being serviced by SQL Server.

The configuration of SQL Server and the operating system may vary depending on whether the workload is transactional or decision-support. If possible, separate these workloads on different Windows instances.

You should also separate production and non-production workloads. As developers test and refine the code to be executed with the database, this unpredictable extra load might degrade the performance of a production system.

The ability of cell-based HP Integrity servers to be partitioned into separate, electrically-isolated Windows instances allows you to combine various types of workloads in a single system.

Only consolidate non-mission-critical workloads first

Since moving into any new environment brings an element of the unknown, you should first consolidate less-critical applications that will not be seriously impacted by downtime. You can apply the lessons learned from consolidating low-risk databases and applications to your future efforts.

In addition, since low-risk databases and applications tend to be less utilized (Tier 3), their consolidation can deliver a faster return on investment (ROI) when underutilized servers are retired or repurposed.

Most development, test, and quality assurance systems are Tier 3; much can be learned from consolidation these environments first.

The partitioning capability of HP Integrity servers allows you to deploy development, test, and quality assurance systems on the same system, which eliminates a variable – an additional server – when you are implementing code changes in a production environment.

Upgrade to SQL Server 2008 before consolidating

While SQL Server 2008 offers an array of features that can be used to enhance a consolidation environment, it may be beneficial to upgrade the database/application to SQL Server 2008 prior to consolidation. This allows you to observe and record the behavior of SQL Server 2008 in a non-consolidation environment, reducing the number of variables and lowering the risk when you ultimately move the workload to a consolidation environment.

Use multiple SQL Server instances when resources permit

While each SQL Server instance consumes server resources, the use of multiple instances helps provide a high degree of isolation. For example, in environments with different workloads (such as transactional versus decision-support), you can use multiple SQL Server instances to separate dissimilar workloads.

The Resource Governor feature of SQL Server 2008 can be used to manage resources for isolated instances.

Note:
When you combine databases from multiple servers, you may occasionally find databases with the same name. If these databases cannot be combined, you should use additional SQL Server instance.

Security concerns over SQL Server administration may provide another reason for deploying additional instances.

Avoid enhancements to applications/databases during consolidation

Scope creep is not only found in software development projects; it can also be found in consolidation projects.

You are already making dramatic changes to the environment during a consolidation project. There may be a platform change from 32-bit to 64-bit processing as well as an upgrade from an earlier SQL Server version and perhaps edition. Any non-essential modifications or enhancements to the databases and applications involved in the migration should be avoided.

Do not collapse many databases into fewer

In some cases, exactly the same databases and database names may reside on many servers across the enterprise. While it makes sense to collapse multiple databases with the same structure and name to a single, consolidated database, this action would, in effect, be an enhancement to an application/database and should thus be avoided.

You should consider consolidating the many databases to fewer after the server consolidation has been successfully completed.

Strive to maintain the user experience when consolidating workloads

One outcome that must be avoided during consolidation is the degradation of the user experience. While consolidation can provide many benefits, if it sacrifices user experience and productivity, the expected ROI may not be realized.

Thorough testing of the consolidation environment can help mitigate this risk and set appropriate expectations.

Drive standardization in the enterprise

Many organizations developed SQL Server sprawl as a result of mergers/acquisitions or a distributed IT strategy. In such an environment, each server in the data center may have its own SLA and its own system and storage configuration, making the environment difficult to manage and support.

As you migrate applications and databases to a consolidation environment, new expectations and SLAs must be established to enhance efficiency and ensure support for databases that may be added to the environment in the future.

Note that SQL Server 2008 can help the consolidation environment maintain SLAs through features like Resource Governor, which, for example, can be used to assign more resources to users under higher level SLAs.

Other caveats

TempDB is another resource constraint that may warrant the creation of another SQL Server instance to distribute the load over additional resources.

TempDB is a universal resource that can be accessed by all the users of a given SQL Server instance; it is used to store objects such as temporary tables. Since features and functionality first introduced with SQL Server 2005 can significantly increase the load on this resource, you should take particular care during the data collection process to size TempDB appropriately.

When planning a consolidation environment, TempDB initial sizing, storage capacity, and throughputs are key concerns. Placing the TempDB data file and log file on separate LUNs can significantly enhance performance.

You must also consider users when consolidating multiple servers and databases: it is possible that the same username may be allocated to two different people on two – or even more – servers. The simple solution would be to give each user in the consolidation environment a unique username; however, this may be impractical due to organizational constraints. As a result, you may have to create separate SQL Server instances, thus reducing the effectiveness of the consolidation effort.

Normally, SQL Server is installed as the default instance on a server, to be used by applications that own that particular server. After being moved to the consolidation environment, a given instance may no longer be the default instance on a server, which can create political rather than technical concerns within the organization. It may be more challenging to change connection strings or connectivity requirements for some applications rather than default instance settings for the target server. In order to deal with this issue effectively, there must be clear executive leadership to resolve political issues.

Impact of additional SQL Server components

This document focuses largely on consolidating the Relational Engine component of SQL Server. However, SQL Server also provides SQL Server Analysis Services (SSAS), SQL Server Integration Services (SSIS), and SQL Server Reporting Services (SSRS). These additional services may be implemented on standalone servers or bundled with the Relational Engine.

This section outlines the impact of SSAS, SSIS, and SSRS on a consolidation initiative.

SSAS

As discussed in the "Guiding principles" section, it is not generally beneficial from a performance perspective to mix workloads. Thus, in general, HP recommends running SSAS on its own server.

Note:
During cube processing, both relational online analytical processing (ROLAP) and online analytical processing (OLAP) consume significant resources. SSAS does not provide a processor affinity mask option.

The following caveats are associated with SSAS:

- SSAS uses two different types of memory: Configuration memory for SSAS and the operating system's file system cache.

- As of SQL Server 2005 SP2, SSAS is NUMA-aware and can take advantage of local memory. For more information on NUMA, refer to the HP Integrity advantage section.
- There is no way to set CPU affinity for SSAS, which limits the possibility of effectively managing CPU resources in a consolidation environment. Thus, if you are consolidating SSAS with the rest of the database engine, a tool such as Windows Server Resource Manager (WSRM) could be helpful for managing CPU resources – but should first be tested in a non-consolidation environment.

SSIS

SSIS is one of the most widely used SQL Server components of SQL Server. Many organizations have relatively simple packages that use SSIS to move or copy small-to-moderate amounts of data, an operation that typically does not cause a major performance hit. However, operations such as cleansing data and loading data into a data warehouse can be very demanding on memory due to activities such as sorting, aggregating, and lookup. If these SSIS operations have a significant impact on server performance, separating SSIS on to its own server may relieve memory pressure but would also increase network traffic. Only by testing in the target consolidation environment can you determine the impact of consolidating SSIS on the same server as the Relational Engine or its impact on the network.

SSRS

SSRS is widely used and will need to be addressed by most organizations planning a consolidation initiative.

There are essentially two layers within SSRS:

- Web layer – The web layer can run on many different servers and point to a common database for reports.
- Database layer – A database layer (ReportServer and ReportServerTempDB) is required for the report catalog. While you do not need a separate SQL Server instance to serve the SSRS database layer, its workload differs from an OLTP workload; thus, it may be beneficial from a performance perspective for SSRS to have its own SQL Server instance.

Popular Microsoft products that utilize SQL Server

Microsoft uses SQL Server as a data store for many of its other products, some of which can create challenges in a consolidation environment. For example, Microsoft Operations Manager (MOM) 2005 can only support a single database per SQL Server instance because the name of the database cannot be changed. MOM 2007 has addressed this limitation.

The popular Microsoft BizTalk Server product has a distinctive workload that requires the MAXDOP parameter to be set, instance-wide, to one. Furthermore, BizTalk Server can only communicate with the default instance of SSAS, not a named instance. Given these limitations, HP does not recommend merging BizTalk Server with another SQL Server instance.

Microsoft Office SharePoint Server, which has grown immensely in popularity, also uses SQL Server to store data. There are no caveats for the use of SharePoint Server in a consolidation environment.

Approaches to consolidation

There are many different approaches to consolidation, each with its own benefits and potential drawbacks, spanning a range from high utilization to high isolation.

This section describes several approaches.

One SQL Server instance per Windows server

Using a single SQL Server instance per Windows server requires no subdivision of resources and is often the most familiar approach to the DBA and other IT resources. With this approach, the server only runs a single SQL Server instance that services all databases. As many resources as possible are dedicated to this instance.

This approach limits the number of SQL Server versions and patches deployed, the size of the attackable surface area, human resource requirements, and more.

While having only one SQL Server instance per Windows server may seem desirable, in practice there is typically a good reason to separate some databases from others.

Multiple SQL Server instances per Windows server

This approach involves the grouping together of similar workloads in conjunction with a few SQL Server instances. The database grouping is based on criteria specified by the particular organization.

Different resource levels can be allocated to each SQL Server instance (using WSRM, for example). Note that SQL Server 2005 and SQL Server 2008 allow processor affinity settings and memory size to be modified without restarting the SQL Server instance.

This approach balances high utilization with isolation capabilities; it can reduce the administrative workload while still providing separation where needed.

One SQL Server instance per subdivided Windows Server

This approach requires a single physical server to be subdivided into smaller physical servers, a capability that is usually found in larger, enterprise-class servers.

Though this approach can essentially collapse SQL Server sprawl into fewer servers, it does not reduce the surface area of versions, editions, and other key management dimensions. It does, however, provide much better separation and isolation of resources without incurring the overhead associated with virtualization.

Subdivision may also lead to low resource utilization since the lowest granularity for processor assignment to a partitioned server is typically four.

HP Integrity servers have the flexibility to combine cells for scalability or keep cells electrically isolated for separation. As business needs change over time, cell-based HP Integrity servers can keep pace.

Virtualized SQL Server environment

Virtualization technologies can be used to consolidate many to fewer physical servers. Each virtual server can provide an environment that supports a Windows instance along with one or more SQL Server instances.

This approach, which can be implemented rapidly if you are just moving from physical to virtual servers, allows you to isolate but still share resources. A major drawback, however, is that you are simply virtualizing SQL Server sprawl, not reducing surface area.

In addition, there is an overhead – often referred to as a performance tax – associated with running a database in a virtual environment. When analyzing a SQL statement, SQL Server develops a query plan and assigns a cost to accessing processor, memory, and disk. These elements are skewed within a virtual environment; thus, a sub-optimal plan may result.

The following virtualization options are available for HP Integrity servers running Windows:

- HP Integrity Virtual Machines (IVM) – IVM uses hardware virtualization and supports guests running with multiple operating system instances on the same host.
- Parallels Virtuozzo Containers (PVC) – PVC uses operating system virtualization, which reduces the performance overhead associated with virtualization.

Although the landscape is changing as next-generation virtualization software, hardware, and database engines are developed, production workloads of any size are not well-suited to the virtual environment. However, non-production workloads such as development, test, and training may benefit from virtualization.

Since SQL Server is not fully supported by all virtualization products, you should investigate and understand support policies before virtualizing a SQL Server environment.

Note:
Some components of SQL OS, a feature of SQL Server 2005 and 2008, interface directly with Windows, taking control of thread scheduling for the processors assigned to SQL Server. SQL OS also communicates to Windows what is happening within the SQL Server engine. To some extent, you can compare SQL OS, a SQL Server instance, to a virtualized component, such as a virtual server.

The above consolidation approaches are general. Many organizations will adopt a hybrid approach since the benefits of one approach will typically fit the needs of one sub-population of servers better than others. For example, you may chose to virtualize databases that are not particularly good candidates for consolidation – for example, they may have low throughput requirements or would free up few resources – but have high isolation requirements. At the same time, you may also employ a single, larger server with multiple SQL Server instances to achieve a balance between utilization and isolation.

Monitoring and managing a consolidation environment

This section outlines a range of approaches for monitoring and managing a consolidation environment – how to implement high availability and disaster recovery solutions, how to manage storage, how to transition a database so that its behavior can be observed, how to enforce standards, and how to use Resource Governor.

High availability and disaster recovery

After the number of servers has been reduced from many to fewer, surviving servers are required to support heavier workloads; thus, their importance grows accordingly. If a single server were to fail in a consolidation environment, multiple databases and users would be affected; prior to consolidation, only a smaller subset of users and applications would have been impacted. As a result, high availability and disaster recovery requirements are often higher in a consolidated environment.

This section outlines how high availability can be achieved through failover and describes options for disaster recovery.

Failover

Microsoft SQL Server 2008 supports the following native automatic failover functionality:

- **Windows Failover Clustering**

  Introduced in SQL Server 7.0, Failover Clustering provides failover at the SQL Server instance level and is well-suited for a consolidation environment.

- **Database mirroring**

  Introduced in SQL Server 2005, database mirroring has been improved in SQL Server 2008 with the addition of compression in SQL Server 2008. Mirroring supports failover at the database level.

  A disadvantage of database mirroring is that mirroring sessions consume resources, leading Microsoft to suggest[3] that no more than 10 databases per instance should be mirrored. Since a consolidation may involve dozens if not 100s or perhaps thousands of databases, mirroring is less attractive in this environment.

Disaster recovery

Native options for disaster recovery include the following:

- **Replication**

  Often difficult to set up, replication is not a true disaster recovery solution; rather, it is method for publishing data to support scaled-out processing. While the recently introduced peer-to-peer replication mode is a better high availability or disaster recovery tool than other modes, it is not an add-on solution for masses of databases; it must be designed into the environment.

- **Database mirroring** (asynchronous mode)

  As described above in the high availability section, database mirroring has inherent limitations that also make it less desirable for disaster recovery. However, if you only need to provide disaster recovery capabilities for a subset of your databases, mirroring may be an option.

- **Log shipping**

  Often used for disaster recovery purposes, log shipping is a time-proven, reliable technology that is fairly simple to set up.

An additional disaster recovery option – one that is not native to SQL Server – has been deployed successfully in many environments and is worthy of consideration. This solution includes the following software components:

- HP StorageWorks Continuous Access – Replicates storage to a disaster recovery location
- HP StorageWorks Cluster Extension[4] – Achieves the failover of SQL Server.

Before going live, you should test this option thoroughly to ensure that it is has been properly configured.

Storage

Storage is a key consideration in a consolidation environment. If the current, non-consolidation environment only features local storage, the migration to a SAN may be a significant change for the DBAs responsible for managing consolidated servers.

Many organizations employ a team whose sole responsibility is to manage enterprise SANs. While the only question this team may ask the DBA focuses on the amount of disk space needed, other factors – such as IOPS and throughput – are just as important.

---

[3] This is not a hard limit.

[4] http://h18006.www1.hp.com/storage/software/replication/index.html

Before the actual consolidation begins, it is crucial that you perform thorough testing to ensure the configuration of the storage subsystem is adequate.

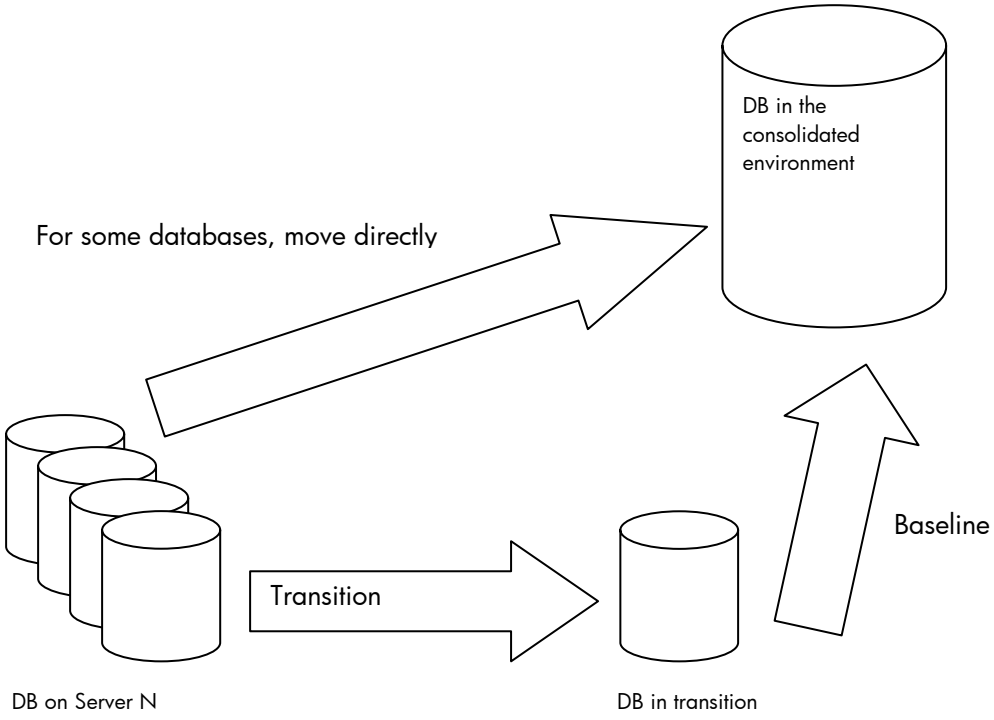
Transitioning a database

Many organizations elect to upgrade their databases to the latest version of SQL Server when consolidating. Whether you choose to upgrade or not, it is often beneficial to evaluate the database on its target platform – featuring the appropriate hardware and database version/edition – in an isolated, non-consolidation environment. By creating a transitional SQL Server instance on the same class of hardware as the target environment, you can observe the behavior of the database(s) associated with a particular application and create a baseline that can be used to determine if there are performance issues when you move to the consolidation environment.

In some cases, moving directly to a consolidation environment without transitioning the database may be a safe course of action. This is especially true when there is no SQL Server version/edition change and the result of the consolidation is merely a hardware change. This typically occurs with databases that have light workloads or databases that have ISV support for a new version/edition of SQL Server. However, if the database is mission-critical in nature and uptime is at a premium, transitioning the database is a sound approach, allowing you to establish a baseline.

Figure 1 outlines your choices.

Figure 1. Transitioning a database



Enforcing standards

Consolidation provides an opportunity to enforce standards – particularly if the IT infrastructure is centralized and there is strong management support – to help cut costs and improve agility.

As servers are retired and databases moved to a new platform, new SLAs that are more consistent with a shared environment can be created. For example, a consistent maintenance window may be enforced on a weekly or monthly basis. This would not require IT to take the system down for a particular period of time every week or month; rather, it sets the expectation that the system might need to be taken down occasionally for maintenance.

Another benefit of consolidation and the enforcement of standards is that, since similar server hardware configurations are leveraged for multiple databases, the performance issues observed with a particular database can be compared against other databases – especially those on the same Windows and SQL Server instances – to help diagnose performance issues. Such observations can aid in diagnosing performance issues as well as configuration differences between multiple SQL Server instances.

SQL Server 2008 introduces support for policy-based management, with capabilities that are somewhat similar to policies in Active Directory. The DBA can establish and enforce policies across multiple servers with multiple SQL Server instances and SQL Server objects of all types (such as Tables, View, and Stored Procedures).

The combination of new SLAs and policy-based management can make the environment easier to manage.

Resource Governor

SQL Server 2008 introduces Resource Manager, a useful feature that can be used to enhance workload management in a consolidation environment. Resource Manager allows resources to be reserved for particular users or workflows based on a user-created function – the classifier – that classifies a connection to SQL Server and associates it with a specific pool of users. Processor resources for this pool may then be throttled as specified.

While a connection cannot be reclassified to a different pool, the resources for the pool may be manipulated.

Resource Governor is engaged when two or more pools are vying for resources. If there is only one pool of users connected, Resource Governor is not engaged; all resources are available for consumption. For example, you could establish a resource pool to run backup jobs overnight; with no other pools connected, the backup jobs would consume the entire processor resource. However, if a user from another resource pool were to connect, Resource Governor would engage and throttle the processing of backup jobs within pre-defined limits.

Each workload or pool can be given an importance level of high, medium, or low, which essentially establishes priorities for resource allocation. In addition, each workload or pool can also be assigned specific operational parameters, including Minimum Memory, Maximum Memory, Minimum CPU, Maximum CPU, and MAXDOP.

Currently, there is a limit of 20 pools or workloads per SQL Server instance.

Resource Governor solves one of the challenges to be addressed when getting different departments to agree to a consolidation effort: that is, assurance that one department's application and database cannot have a significant impact on the performance of another's. The use of Resource Governor helps provide a predictable response and limits the adverse affects of a runaway query.

HP Integrity advantage

The HP Integrity line of Itanium-based servers scales from two-processor blades to 64-processor – or 128-core – Superdomes. For most consolidation scenarios, HP Integrity models that provide the most benefit start at the mid-range (rx76xx) and go up to rx86xx and Superdome models. The main reason for recommending these particular models is their scalability.

The Itanium processor is built to support true 64-bit processing; it includes an instruction set that has been designed for scalability. Working closely with Intel on the development of the Itanium processor, HP has also created an efficient chipset designed specifically for this CPU.

While memory can be a limiting resource in some servers, with mid-range HP Integrity servers and higher, HP employs a cell-board based architecture that implements NUMA for memory access. Each cell board has its own set of four processor sockets, memory slots, and I/O bus. Cell boards are interconnected using cross-bar technology.

Applications such as SQL Server 2005/2008 that are NUMA-aware understand the concepts of near memory (using memory on the same cell board as the processor that is executing an instruction) and far memory (using memory on a different cell board). Since memory access on the same cell board as the processor is much faster, SQL Server 2005/2008 uses this information when scheduling processor threads to make memory access – and therefore processing – faster. Applications that are not NUMA-aware do not differentiate between near and far memory and, thus, cannot optimize thread scheduling.

I/O is one of the first resources to be investigated by DBAs and other IT professionals when performance problems arise. This is for good reason; I/O is often the slowest component of a system.

Although the HP Integrity line of servers can address a significant amount of memory, the amount of data in today's databases – especially those with regulatory retention requirements – is growing rapidly. Since most servers no longer have the capacity to cache an entire database in memory, there is always data on the storage medium that needs to be read into the server or written back out.

The I/O subsystem in HP Integrity servers has an advantage in that each processor has a dedicated PCI bus slot available for use as a direct conduit to the storage subsystem. This approach is intended to avoid traditional bottlenecks and promote an architecture where the addition of resources results in a near-linear performance increase.

Scalability

Scalability is a difficult concept to define. Wikipedia, for example, defines scalability as "a desirable property of a system, a network, or a process, which indicates its ability to either handle growing amounts of work in a graceful manner, or to be readily enlarged."[5] Werner Vogels (CTO, Amazon.com) has suggested that[6] "[a] service is said to be scalable if when we increase the resources in a system, it results in increased performance in a manner proportional to resources added."

The HP Integrity line of servers has been designed to provide scalability. If you expect your workload to double in size, you should be able to maintain satisfactory performance by doubling your processor resources[7].

---

[5] Accessed November 2008

[6] http://www.allthingsdistributed.com/2006/03/a_word_on_scalability.html (accessed October 2008)

[7] Testing performed by HP has demonstrated that SQL Server scalability on HP Integrity servers is near-linear. As with any laboratory testing, however, these results are idealized; in a production environment, scalability may be impacted by a variety of factors. HP recommends proof-of-concept testing in a non-production environment using the actual target application as a matter of best practice for all application deployments. Testing the actual target application in a test/staging environment identical to, but isolated from, the production environment is the most effective way to characterize system behavior.

Consolidation platform

With its ability to run SQL Server 2000 natively, an HP Integrity server can provide a solid platform for applications that cannot take advantage of SQL Server 2008 but may benefit from consolidation.

For those applications that can move up to SQL Server 2005 or SQL Server 2008, the HP Integrity server offers a true 64-bit computing platform for your most demanding workloads, with support for a high degree of consolidation.

The entry-level HP Integrity rx6600 Integrity server can deliver the I/O capabilities and memory capacity needed to support basic consolidation efforts. Mid-tier Integrity rx7640 and rx8640 servers provide robust hardware platforms that offer flexibility and scalability, while the powerful Superdome delivers the most flexible, reliable, scalable consolidation platform.

Performance testing

In July 2008, Scalability Experts utilized HP facilities to characterize the performance of an HP Integrity rx8640 server running SQL Server 2008 Release Candidate 0 (RC0). The objectives of this effort were as follows:

- Evaluate the scalability of a server running a real-world workload
- Establish a baseline for consolidation by isolating a second workload on the tested server
- Evaluate the performance of the tested server after consolidation, running both workloads

Based on their wide experience with many customers, Scalability Experts was able to provide two real-world workloads – Workload A and Workload B – for this test effort. These workloads originated from different online transaction processing (OLTP) systems.

Initially, Workload A was used to characterize system performance as additional resources (processor and memory) were added to the tested server. To simulate a consolidation environment, Workload B was then added and the effects of increasing the total load on the system observed.

Scalability test using Workload A

Workload A was tested with a range of system resources. The number of processors and amount of memory available to SQL Server 2008 were governed by affinity settings.

The following test scenarios were explored:

- Test 1 – Two dual-core Itanium processors (cores 0 – 3); 32 GB RAM
- Test 2 – Four dual-core Itanium processors (cores 0 – 7); 64 GB RAM
- Test 3 – Six dual-core Itanium processors (cores 0 – 11); 96 GB RAM
- Test 4 – Eight dual-core Itanium processors (cores 0 – 15); 128 GB RAM

The workload was replayed at full speed utilizing SQLscaler 2.5, a product developed by Scalability Experts and marketed by Idera.

The only other activity performed by the test system was Perfmon data collection. Key counter values are shown in Table 1.

Table 1. Perfmon counter values for Workload A

Counter (average)	Test 1	**Test 2**	**Test 3**	**Test 4**
Page Faults/sec	461	411	408	387
SQL Compiles	8.3	12.6	100	231
Context Switches/sec per core	992.5	509.375	399.16667	298.375

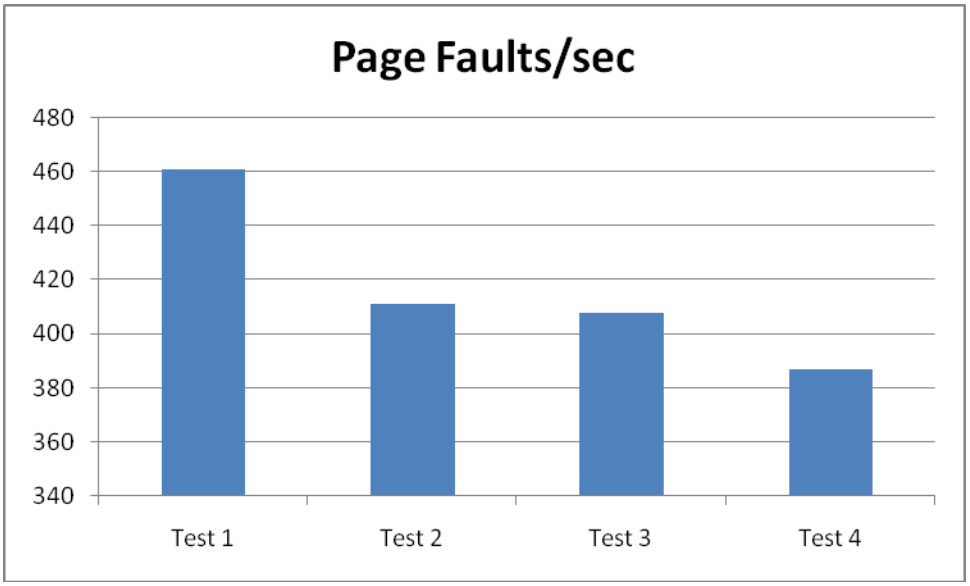
Specific counter values are discussed below.

Page Faults/sec

The Page Faults/sec counter indicates how often the system had to go to disk to retrieve the information necessary to complete an action. The value includes both hard and soft faults.

Although the Page Faults/sec value does not by itself tell the complete story, Figure 2 shows that the number of faults tended to decrease in a near-linear fashion as resources were increased.

Figure 2. Page Faults/sec values for Workload A



Note:
There are slight plateaus in Page Faults/sec values as cell board boundaries are crossed.

SQL Compiles/sec

SQL Compiles/sec values, shown in Figure 3, provide an indication of processing rates.

Figure 3. SQL Compiles/sec values for Workload A

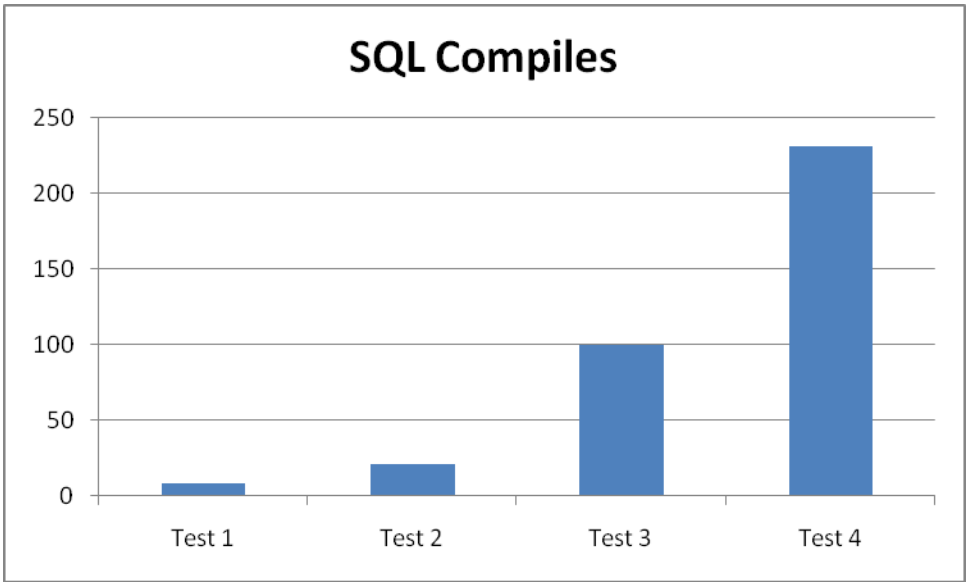


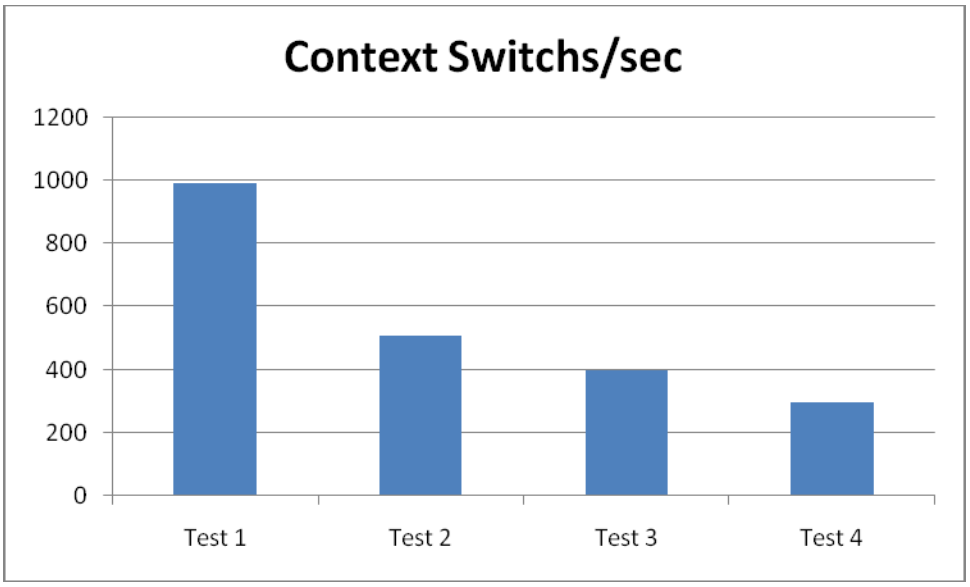
Figure 3 shows processing capabilities climbing as processors were added.

Context Switches/sec

Context Switches/sec values, shown in Figure 4, show how many times per second the processors switched their processing contexts from privileged (operating system) activities to unprivileged (application – in this case, SQL Server) activities.

Higher values indicate less efficient processing.

Figure 4. Context Switches/sec values for Workload A



Although the values shown in Figure 4 are very satisfactory, they show how little pressure was exerted on each processor core.

Workload B isolation test

Before adding it to a consolidation environment, you should first isolate a workload to determine if any bottlenecks exist. If so, these bottlenecks may affect overall performance in a consolidation environment.

Thus, to establish a baseline, Workload B was run in isolation on the test system using the Test 1 scenario (two dual-core Itanium processors, 32 GB RAM). Workload B was similar to Workload A in that it was an OLTP workload; however, it originated from a different customer.

Table 2 outlines the test results.

Table 2. Perfmon counter values for Workload B in isolation

Counter (average)	**Test 1**
Page Faults/sec	394
SQL Compiles	504
Context Switches/sec per core	1545

Table 2 reveals that, compared to Workload A, Workload B exhibited lower Page Faults/sec values, much higher SQL Compiles/sec values, and approximately 50% higher Context Switches/sec values, suggesting that Workload B was more computationally heavy, with memory accesses occurring at a lower rate.

Consolidation test – combined workloads

During the final phase of testing, both Workloads A and B were replayed using the Test 4 scenario (eight dual-core Itanium processors, 128 GB RAM).

Because combined workloads sometimes interact with each other in different ways in different portions of the system, overall performance metrics may be unexpected.

Table 3 outlines the test results.

Table 3. Perfmon counter values for combined workloads

Counter (average)	**Test 4**
Page Faults/sec	482
SQL Compiles	445
Context Switches/sec per core	372

Figures 5 – 7 compare system performance with Workload A (Test 1 – 4) and with the combined workload (Test 4).

Page Faults/sec

Figure 5. Comparing Page Faults/sec values for Workload A and the combined workload

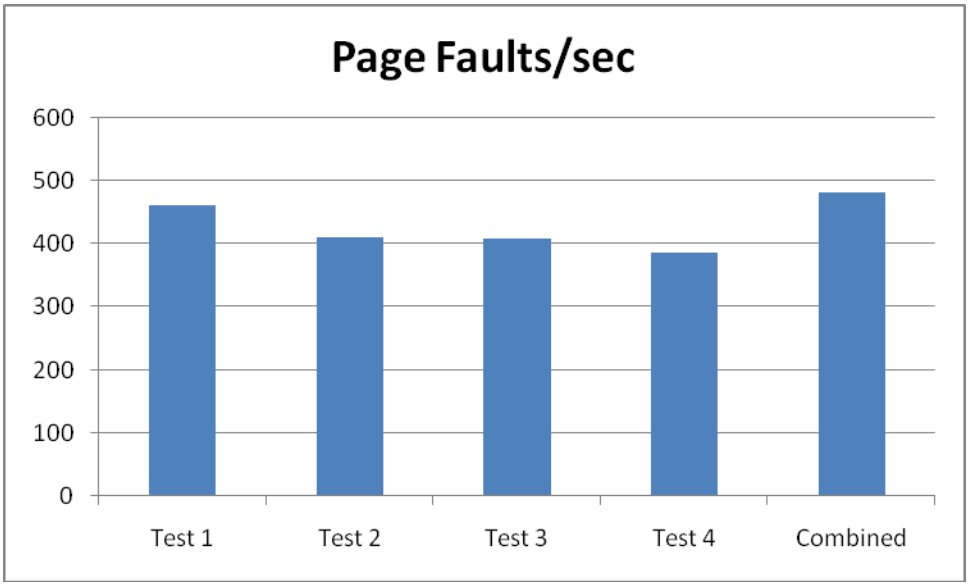


Figure 5 shows that, as expected, Page Faults/sec with the combined workload increased. With the same configuration (eight dual-core Itanium processors, 128 GB RAM), the value for Workload A was 387, approximately 20% less.

Note that with Workload B (two dual-core Itanium processors, 32 GB RAM) there were 394 Page Faults/sec, demonstrating that you cannot estimate system performance in a consolidation environment by adding together individual metrics from non-consolidation environments. Other factors, such as storage system configuration, number of users, and relative costs of the plans in cache, can significantly impact performance comparisons.

SQL Compiles/sec

Figure 6. Comparing SQL Compiles/sec values for Workload A and the combined workload

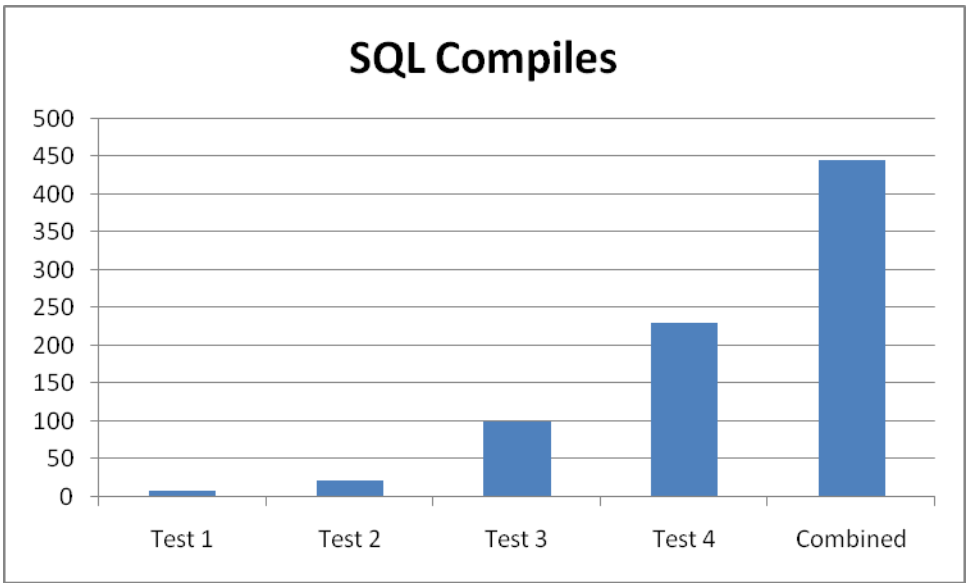
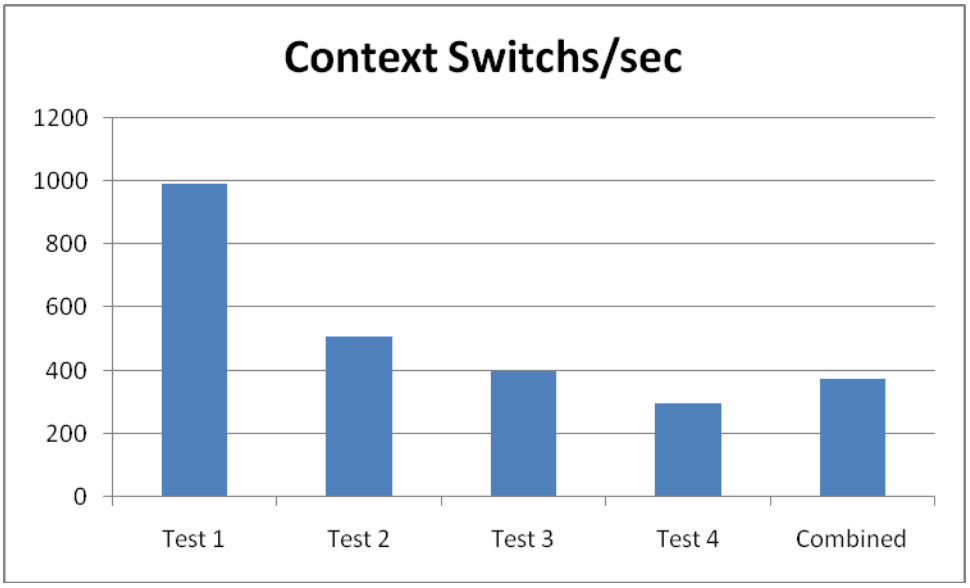


Figure 6 indicates that the system could easily accommodate the additional workload (Workload B), which was computationally heavier than Workload A.

Figure 7. Comparing Context Switches/sec values for Workload A and the combined workload



Again, Context Switches/sec counter values are very good; the additional load only has a slight impact on context switching.

Thus, the consolidation of workloads on the HP Integrity rx8640 server running SQL Server 2008 RC0 was accomplished satisfactorily. By carrying out baseline performance tests while the second workload was isolated, test personnel were able to determine and account for the characteristics of this workload.

Summary

SQL Server 2008 provides features that make SQL Server consolidation even more attractive. These features include:

- Resource Governor makes it easier to manage workloads
- Encryption adds another layer of security
- Compression frees up valuable disk space without sacrificing performance
- Policy-based management allows the DBA to create policies for the enforcement of enterprise-wide standards

There are many ways to approach consolidation; you should analyze the advantages and disadvantages of each before moving forward. Combining the SQL Server 2008 database platform with reliable, scalable HP Integrity servers can make any consolidation effort more likely to succeed.

A key benefit of using HP Integrity servers is their flexibility, which allows them to adapt to changing business needs. These enterprise-class servers may be partitioned into electrically-isolated portions and can thus run SQL Server 2000, 2005, and 2008 on the same platform. In addition, HP Integrity servers can take advantage of the large, flat virtual address space that 64-bit computing affords SQL Server databases.

For more information

HP Integrity servers	www.hp.com/go/integrity
HP storage	www.hp.com/storage
HP and Microsoft	www.hp.com/go/microsoft
Microsoft SQL Server	http://www.microsoft.com/sqlserver/2008/en/us/default.aspx
HP Solutions with Microsoft SQL Server	http://h71028.www7.hp.com/enterprise/cache/3887-0-0-0-121.html
Windows on HP Integrity Servers	http://h20341.www2.hp.com/integrity/cache/497701-0-0-0-121.html?jumpid=reg_R1002_USEN

To help us improve our documents, please provide feedback at www.hp.com/solutions/feedback.

4AA2-3419ENW, November 2008