Windows IT Pro

How to Guide: SQL™ Server 2005 Consolidation

By Randy Dyess

Edited with permission from *SQL Server Magazine*.

How to Guide: SQL Server 2005 Clustering

Contents

Background on SQL Server Consolidation: What Is It?

As organizations around the world increase the number of applications that collect and utilize data, they are increasing the number of servers that must be used for both the collection and storage of that data. Data centers designed to hold dozens of servers are now holding hundreds. System administrators have grown in number and are faced with the problem of running these hundreds of servers in data centers with inadequate power and inadequate cooling.

As these administrators face this surge of new hardware, management faces the ever increasing costs of running the data centers. Power costs, hardware costs, and personnel costs are rising and straining budgets. To overcome this budget-bursting surge of equipment and manpower, server manufacturers have created larger and larger servers capable of handling the workload of multiple smaller servers. System administrators and management are now finding that a new process—consolidation—has come into place that will allow the replacement of multiple, smaller machines with larger more powerful servers.

Consolidation is the grouping of multiple applications or databases on one server. Whether this consolidation takes place through the virtualization of servers—multiple servers running independently on a host server—or the consolidation is designed to have all applications or databases under the same operating system, consolidation allows the use of one larger machine in the place of multiple machines and helps data centers reduce the number of servers they have to support.

First things first: Why would I consider consolidation?

There are many reasons for consolidation, both technical and non-technical, that you should understand as you consider consolidation in your database environment. These reasons can be broken down into three major sections: Flexibility, Manageability, and Performance.

Flexibility

A SQL Server consolidation solution has many different architectural designs: Multiple databases being consolidated onto a single SQL Server instance; multiple instances of SQL Server being consolidated onto one physical server; multiple virtual servers on a single physical server. Whichever design you choose, and you may choose multiple designs in your environment, the flexibility that comes with each different environment allows you to choose the degree of isolation you require based on application or business requirements.

Achieving flexibility in your consolidation architectural design involves multiple items you must consider when planning your consolidated environment. You will need to consider items ranging from the resources of the hardware to the security of your consolidated environment.

Hardware Resources

Each different consolidation architectural design model has a different effect on how the resources of your physical server will be utilized. You will need to understand how the design affects memory, CPU, and the disk subsystem to decide if you need a 32-bit or 64-bit environment and how you can divide the resources on each system to meet the needs of your design.

Workload Isolation

When planning your design, you need to consider any requirement for isolating your workloads. If you have business requirements that certain workloads must be isolated due to security or performance concerns, you will need to choose the correct architectural design for these requirements. Often, administrators will find themselves with a mixed consolidation environment in order to isolate the workload of an application due to the performance of that application and the need to ensure that the application's performance is not affected by other database workloads.

Application Compatibility

As with workload isolation, often you will need to isolate applications on different consolidated servers due to the configuration options, access protocols, or collations requirement by the different applications. While you can place multiple databases with different collations into one SQL Server instance, it is often better for ease of administration to install different SQL Server instances with different collations.

Availability

The design of your consolidated environment can be determined by the high availability needs of your different databases and applications. You would not want to place a database that is mission-critical and is required to be up with dozens of other databases that are not mission-critical and often are overlooked during database maintenance. These non mission-critical databases may create problems that cause the mission-critical database to experience more than the number of outages permitted by the SLA of that database. The need for certain databases to always be available will determine how you place these databases in your architectural design.

Security

Consolidating many different databases onto one server leads to the question of security and how you configure your security to allow for the administration and management of those databases by different administrators. Do you have one instance with many different DBAs in the sysadmin group? Do you have multiple administrators placed into the administrators group on the server? Do you need to separate the security requirements for different applications and their databases or does one group handle all databases and therefore you are allowed to create one group to handle all the databases on one consolidated server.

What about configuration settings? Does one application require the ability to have SQLCLR in its database? Will you allow this option for all the databases in the instance or will you need to separate that application's database into a different SQL Server instance? The many different security needs of consolidated databases will often be a factor in the design of your consolidated environment.

Manageability

An issue often overlooked in data centers is the large amount of manageability that is needed to administrate multiple servers. Consolidating servers and databases reduces the amount of workload to administer your database environments because it reduces the number of servers you have to administer. Instead of having to worry about the management of multiple servers when each new hotfix comes out, you only have to worry about the management of a few servers.

Monitoring and auditing are reduced in consolidated environments. No longer do administrators have to worry about the monitoring needs of hundreds of servers. No longer do administrators have to have methods for consuming all the monitoring and auditing data that comes from monitoring hundreds of servers. Consolidating your databases into as few physical servers as possible that will support your throughput requirements often means that your database staff can move from the day-to-day "fire fighting" mode that many database groups live their lives by to a more pro-active day-to-day workload that gives them time to solve issues before they become a problem or to simply have time to address performance design and configurations that often lead to even better utilization of your system resources.

Performance

A question that often comes up when considering consolidation is how can you manage the performance of your databases and servers after you move multiple databases and applications from many different servers, each with their own resources, to a few servers and then having multiple applications share the same system resources? Performance of your databases and applications is a major issue when determining if and how you are going to consolidate your database environment.

Understanding the performance needs of your individual databases and the applications that utilize those databases are the key to understanding how you will manage those resources after you consolidate your databases. Setting up a performance monitoring solution before consolidation will allow you to match the needs of your database's resources to the needs of the other databases being consolidated on a single physical server. Often you will find yourself dividing your databases into different classes based on their performance needs and then assigning databases to servers based on their classes and how many different databases you have already assigned to that server at the same or at a higher performance class.

Another component of performance to consider is how different the system resource usage will be on many smaller 32-bit servers versus one larger 64-bit server. Often the inability of SQL Server to fully utilized AWE memory on a 32-bit installation for memory other than data allows the consolidation of many 32-bit installations onto on 64-bit install actually increases the performance of those databases instead of decreasing them due to their need for larger memory area given to procedure cache.

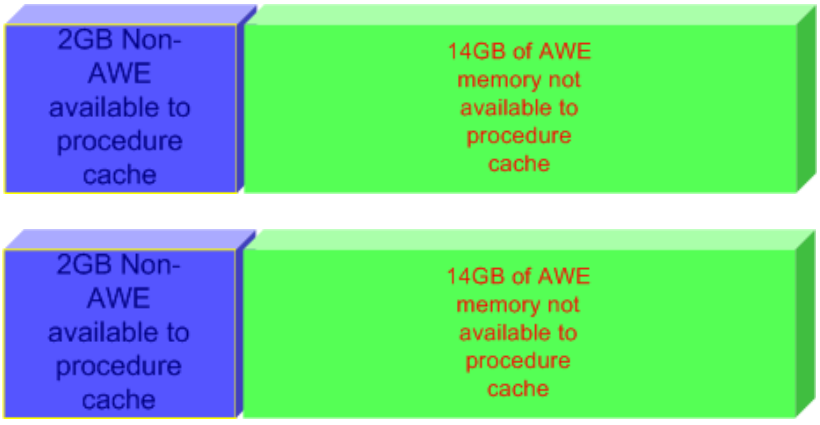

Figure 1 ***Restrictions of procedure cache on 32-bit installation***

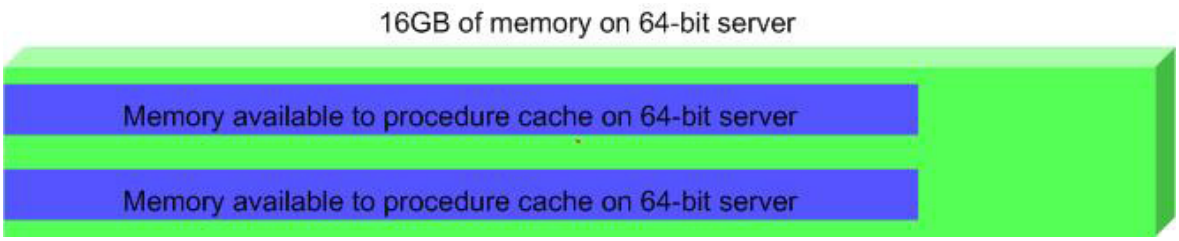

Figure 2 ***Larger amount of memory for procedure cache on 64-bit installation***

Simple Steps to Getting Started

As a database administrator, you may be asking yourself how you would get started with consolidation. Consolidating SQL Server databases involves planning. This planning involves multiple stages: determining which database instances to consolidate, determining the method of consolidation, determining the performance metrics that must be met after consolidation, and planning the movement of the databases during the consolidation.

Which databases should I consolidate?

When you start your consolidation planning, you will need to determine which databases to consolidate. As stated in the sections above, there are many different considerations which you must take into account before consolidating your databases. Often database administrators are tasked with deciding which of their databases to consolidate and which ones to leave on separate physical servers. Database administrators should evaluate each database separately to determine if it is a good candidate for consolidation. You shouldn't plan on consolidating every database to one large server. This method often leaves you with undesirable performance after moving too many databases with large resource utilization to one server, even if that server is much larger than you are accustomed too.

What performance metrics should I capture before and after consolidation?

Capturing a basic set of performance metrics is critical before you consolidate in order to understand how your databases utilize system resources. Understanding system resource usage can be achieved with standard performance monitoring tools found with Windows® and SQL Server: Windows System Monitor (Sysmon or Perfmon), SQL Server Profiler, and SQL Server dynamic management views/functions.

Perfmon memory counters should include:
- Available Mbytes
- Free System Page Table Entries
- Page Faults/sec
- Pages/sec
- Pool Nonpaged Bytes
- Pool Paged Bytes
- Transition Faults/sec
- Buffer Manager:Buffer cache hit ratio
- Buffer Manager:Buffer cache hit ratio
- Buffer Manager:Checkpoint pages/sec
- Buffer Manager:Free pages
- Buffer Manager:Lazy writes/sec
- Buffer Manager:Page life expectancy
- Buffer Manager:Procedure cache pages
- Buffer Manager:Readahead pages/sec
- Cache Manager(_Total):Cache Hit Ratio
- Cache Manager(_Total):Cache Object Counts
- Cache Manager(_Total):Cache Pages
- Memory Manager:Memory Grants Pending
- Memory Manager:Optimizer Memory (KB)
- Memory Manager:SQL Cache Memory (KB)
- Memory Manager:Target Server Memory(KB)
- Memory Manager:Total Server Memory (KB)
- Paging File(_Total):% Usage

Perfmon CPU counters should include:
- Process(sqlservr):Page Faults/sec
- Process(sqlservr):Private Bytes
- Process(sqlservr):Virtual Bytes
- Processor(*):% Privileged Time
- Processor(*):% Processor Time
- Server Work Queues(*):Queue Length
- Server Work Queues(Blocking Queue):Queue Length
- System:Context Switches/sec
- System:Processor Queue Length

Perfmon I/O counters should include:
- LogicalDisk(*):% Disk Read Time
- LogicalDisk(*):% Disk Write Time
- LogicalDisk(*):Avg. Disk Queue Length
- LogicalDisk(*):Avg. Disk Read Queue Length
- LogicalDisk(*):Avg. Disk Write Queue Length
- LogicalDisk(*):Disk Bytes/sec
- PhysicalDisk(*):% Disk Time
- PhysicalDisk(*):Avg. Disk Bytes/Read
- PhysicalDisk(*):Avg. Disk Bytes/Write
- PhysicalDisk(*):Avg. Disk Queue Length

Additional Perfmon counters should include:
- Server:Logon Total
- Access Methods:Full Scans/sec
- Access Methods:Page Splits/sec
- Access Methods:Table Lock Escalations/sec
- Access Methods:Worktables Created/sec
- Databases(*):Log Flush Wait Time
- Databases(_Total):Log Flush Waits/sec
- Databases(_Total):Transactions/sec
- Databases(tempdb):Transactions/sec
- General Statistics:Logins/sec
- General Statistics:Logouts/sec
- General Statistics:User Connections"
- Latches:Average Latch Wait Time (ms)
- Latches:Latch Waits/sec
- Latches:Total Latch Wait Time (ms)
- Locks(*):*

- SQL Statistics:Auto-Param Attempts/sec
- SQL Statistics:Batch Requests/sec
- SQL Statistics:SQL Compilations/sec
- SQL Statistics:SQL Re-Compilations/sec

You should run Profiler traces capturing the following events: RPC:Completed, RPC:Started, SQL:BatchCompleted, SQL:BatchStarting, Lock:Deadlock, Lock:Timeout, SP:Recompile, SQL:StmtStarting, SQL:StmtCompleted, SP:Starting, and SP:Completed. If you are familiar with the new SQL Server 2005 Dynamic Management Views and Functions, you can utilize them to capture much of the same information as you do using the SQL Server Profiler. Dynamic Management Views that are useful include: all sys.dm_exec, all sys.dm_os_memory, and sys.dm_os_wait_stats.

How should I move databases during the consolidation phase?

Fortunately, the methods you need to move your databases during the consolidation phase are the same methods we utilize when migrating our databases during side-by-side upgrades onto different servers. In fact, the many whitepapers and articles that discuss upgrading your databases from one version to another are a perfect source of information on how to move your current environments from many different servers to one consolidated server.

Detach/Attach

When moving a database from one server to another during a consolidation phase many database administrators choose to simply detach the database from the source server and reattach it onto the consolidated server. This method is often easy to accomplish because it requires very few steps to detach the database, move the database files to the consolidated server, and then reattach the database. The issues faced with choosing the method is the downtime associated with detaching the database since you will not be able to allow users in the database while it is being detached and while you are copying the detached file to the consolidation server.

Backup/Restore

Another common method used to move a database from one location to another is the use of a current backup file to restore the database on the second server. Utilizing this method requires that a full backup be taken and then the backup file copied to the second server where it is restored. The database must be restored without being recovered if there are still active connections taking place in the source database. Once the full backup is restored, in order to synchronize your databases after making the backup you will need to stop all activity in the source, create a transaction log backup and then apply the transaction log backup onto the database on the consolidated server while recovering the database. Once you have done this, you will have two databases that are synchronized and you can either reconnect all applications to the new consolidated database server or create a data movement method such as transactional replication to keep the data synchronized until all applications are configured to connect to the consolidated server.

Database Mirroring

With the inclusion of database mirroring in SQL Server 2005, database administrators have a powerful new method to move databases to a consolidated server. Database mirroring maintains two copies of a single database that must reside on different server instances of the SQL Server Database Engine. Typically, these server instances reside on computers in different locations. One server instance serves the database to clients (the principal server). The other instance acts as a hot or warm standby server (the mirror server), depending on the configuration and state of the mirroring session. When a database mirroring session is synchronized, database mirroring provides a hot standby server that supports rapid failover without a loss of data from committed transactions. When the session is not synchronized, the mirror server is typically available as a warm standby server (with possible data loss).

Database mirroring allows you to set up a mirror of a database on a different server, in our case the consolidated server, and the mirroring mechanism will keep the data between the principal database, the source database, and the mirror database, the consolidated server database. Once we have mirroring configured, it is a simple matter of failing over the mirrored environment. This failover will configure the mirror server as the new principal, reconnect many applications automatically to the new principal, and make the current principal into the mirror. The mirroring mechanism will now keep the database on the old server current until you are satisfied that you have configured all your applications to utilize the consolidated server database. After you have decided that you are comfortable with the consolidated server, you can simply break the mirroring and remove the old database from the old server.

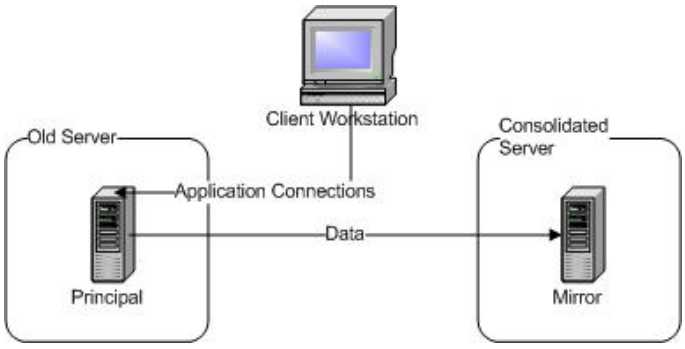

Figure 3 *Mirroring before failover to consolidated server*

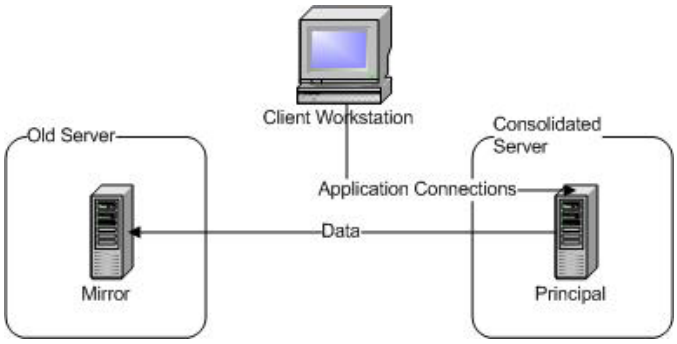

Figure 4 *Mirroring after failover to consolidated server*

How do I evaluate my SQL environment?

Often when a company is considering consolidation of their databases, the question arises of how to evaluate the current SQL Server environments before consolidation. Evaluating current servers can take place on multiple levels: how critical is the database, the performance and resource needs of the database, the size and structure of the database, and what special configurations are needed for the database at its applications.

Criticality of the Database

Often the determining factor when deciding to consolidate databases is the criticality of the database. Database deemed mission-critical are often left on individual servers while databases that are considered less critical or non-critical are considered as candidates for consolidation. A consolidation scheme which places many non-critical on one physical server or utilizes a layering of critical and non-critical databases on a larger server in a fashion that bests utilizes the resources of that server without comprising the criticality of the most mission-critical databases.

When determining the criticality of databases many database administrators utilize a tier structure. Tiers 1, 2, 3, and sometimes 4 are assigned to different databases and applications based on the criticality of the database and application. Tier 1 databases are often the largest, most heavily used databases that are essential to the company's operations and revenue. These databases and applications often have Service Level Agreements (SLA) that require minimal yearly downtime. Tier 1 databases are often left on individual servers in order to prevent operations and resource usage of other databases from interfering with their performance and availability.

Tier 2 databases are critical but can often undergo small amounts of downtime or performance issues without violating SLAs. These databases are often place together in small groupings on consolidated servers based on their workload types and resource usage. Tier 2 databases are a great candidate to be consolidated with multiple resource light tier 3 and 4 databases.

Tier 3 and 4 database are the multiple smaller and lightly used databases that can often undergo long periods of outages or performance issues without affecting the operations and revenue of the company. These databases often include databases that are used for reporting that are not needed in real-time and can be re-executed after the database is brought back online after a failure without loss of data in the report. These databases can also be tiered based on their system resource usage. A database that may be critical but is very small and does not utilize a large amount of resources can be labeled a tier 3 database while databases that can undergo large amounts of downtime can be labeled tier 4.

Whichever method you use to level your databases, the use of tiers for databases being considered for consolidation is a good way to start your initial consolidation planning.

Performance Needs

Performance needs of a database are always one of the primary areas to use when evaluating your databases for consolidation. Often database administrators will need to establish performance baselines for a period of time before consolidating in order to actually understand the needs of the database. Databases that have large performance needs in one or more resource areas: memory, CPU, and disk subsystem should not be consolidated together if they are consolidated at all. As with criticality of the databases, tier levels can be determined for each

level of resource usage and database assigned to consolidation servers based on their tier levels. However the database administrator decides on leveling the database and however they decide on placing database on different consolidation servers, database administrators must consider the total resource needs of all potential consolidated databases in order to ensure a successful consolidation project.

Database Size and Structure
While databases can be consolidated for a variety of needs, databases that are very large or configured with a special structure (large use of files and filegroups) should often be placed on the "do not consolidate" list. Database size and the configuration of the files and filegroups should be on the list of items to evaluate when choosing which databases are to be consolidated. Consolidation not only places strain on the memory and CPU of the consolidated server, it places strain on the disk subsystem and database which require large amounts of disk space or large numbers of drive letters for its files and filegroups may not be good candidates for consolidation with other databases.

Special Configurations
Databases needing special configurations, such as legacy network protocols or legacy components, may not be good candidates for consolidation with databases that do not require the same legacy components. Often applications need older versions of components that do not exist with the latest release of the component. These special needs must be addressed when evaluating databases for consolidation. Configuration needs such as different service packs or configurations in the operating system may mean the application and database can't be consolidated without the use of a virtualization product to separate the configurations from other databases and applications on the consolidated server. If you are consolidating using virtualization products, then the special configuration needs are not an issue, but if you are not using virtualization products for your consolidation then you must include these needs in your evaluation.

Common examples and reason for SQL Consolidation

There are many common examples and reasons for consolidating SQL Server but one that seems to be the most common is consolidation for the reduction of physical servers. Many data centers are filling up and this causes issues with both the cooling of the centers and the need for more energy to be brought into the center. Many data centers, especially those in California, are facing critical needs in trying to figure out how to bring additional energy into the data center for the large amounts of servers that are housed there. Consolidating servers will reduce the amount of energy needed for data centers and often this energy reduction will not only reduce the need to create additional data centers, it will reduce the budget needed to run the data center.

Best Practices to Getting Started

Like anything else, there are a few best practices that database administrators can follow to start the consolidation process:

- Obtain the largest servers you can afford for the consolidated server
- Install as much memory on the consolidated server as you can afford
- Configure your disk subsystem so each instance receives enough drives for its system databases, tempdb, transaction logs, and user database data files

Do's and Dont's when Consolidating Databases

When consolidating databases, database administrators will need to remember a few do's and don'ts when planning their consolidation and undergoing their consolidation phase:

- Do not consolidate large, mission-critical databases
- Do not consolidate more than two Tier 1 databases onto the same server
- Do not consolidate more than 10 instances onto one server
- Do understand the performance characteristics of your databases and match those to the consolidation server
- Do not mix different workloads into one consolidated instance
- Do consolidate 32-bit installations onto 64-bit installations when you have the choice
- Do test your consolidation strategy before moving databases
- Do not consolidate more than 500 databases into one instance

Common Questions and Challenges

The common questions and challenges that are often raised during consolidation projects usually centered on the labeling of databases according to tier levels. It is during this evaluation phase that database administrators often have to figure out the performance metrics, database configuration metrics, database criticality, and business service level agreements that may leave more questions unanswered than answered. Questions arise from

what metrics to capture, how configurations will affect the consolidated environment, and how database criticality can be maintained after a consolidation. When you start a consolidation project the number one item you can do is to understand your database environments as much as you can before starting the consolidation project.

Ready to get started?

If you are a database administrator considering consolidating databases, you will need to create a checklist that will allow you to know which databases you will be consolidating and how you will match those databases to the proper consolidated server. You will also need to create a performance metric analysis document that describes the performance metrics you captured, what those metrics were before the consolidation, what the metrics are after consolidation, and how the two sets of metrics differ. The next part of your initial planning will be to consider each data movement method and determine which method is appropriate for each database. Once you complete each of these checklists, you are ready to begin your consolidation planning and eventually, your consolidation.

For more information, contact Dell. Information in this document is subject to change without notice.

Randy Dyess, Solid Quality Learning Mentor and Program Manager: Strategic Initiatives, has a variety of experiences dealing with SQL Server over the past nine years and has worked with environments with Terabytes of data and environments that had over 1,000 databases with only a few megabytes of data in each database. Currently, Randy is the founder and owner of Dyess Consulting Inc. a SQL Server mentoring and training consulting firm which specializes in training and mentoring in Transact-SQL and SQL Server performance tuning and database security. Randy is the author of TransactSQL Language Reference Guide and numerous magazine and newsletter articles pertaining to SQL Server security and optimization issues and has spoken at various international and national conferences.