SQL Server Consolidation Guidance

SQL Server Technical Article

Writers: Sung Hsueh, Antony Zhong, Madhan Arumugam

Technical Reviewers: Claude Lorenson, Clifford Dibble, Lindsey Allen, Sambit Samal, Sethu Kalavakur, Prem Mehra, Sameer Tejani, Il-Sung Lee, Jack Richins, Brian Dewey, Mathew John, Jamie Reding, Jonathan Morrison, Omri Bahat, S Muralidhar, Haydn Richardson

Editor: Beth Inghram

Published: November 2009

Applies to: SQL Server 2008 and SQL Server 2008 R2

Summary: The goal of this white paper is to provide a framework for choosing among virtualization, multi-database, and multi-instance consolidation strategies for SQL Server Database Engine OLTP applications by highlighting some of the key decision points based on technical analysis. Some foundational topics and terminology will be included to provide the basis for the discussion, because some terms or strategies might be worded differently in other papers.

Introduction

Consolidation, in general terms, is the combining of various units into more efficient and stable larger units. When applied to an IT department, consolidation specifically translates into improved cost efficiency from higher utilization of resources, standardization and improved manageability of the IT environment, and (more recently) a focus on a "green" IT environment through reduced energy consumption. One of the important components in the IT environment is the database. Databases tend to be very widespread across the enterprise, because they are very efficient for handling relational storage of data and are designed to be platforms for a wide variety of applications. Because databases form the foundation of so many business systems, IT departments can easily lose control of the number of databases that need to be maintained, because each group may simply create their own database to solve a specific problem they may be experiencing. This leads to a proliferation of databases and machines running database instances also known as *database sprawl*. Thus databases are one of the prime candidates for consolidation. When consolidating database applications, consider the following three potential strategies: using a single physical machine to host multiple virtual machines (VMs) running Microsoft® SQL Server® data management software, using a single machine to host multiple SQL Server instances, and using a single instance of SQL Server to host multiple databases. Each of these strategies has different advantages and disadvantages related to security and compliance requirements, high availability and disaster recovery requirements, resource management benefits, level of consolidation density, and manageability tradeoffs.

This paper will try to answer the following questions:

- What are the considerations when creating a consolidation plan for my environment?
- What are the key differentiators among the three consolidation options?
- How can I use these differentiators to choose the appropriate consolidation option for my environment?

Based on our experiences and the feedback we have had from customers and partners, we will try to answer these questions by:

- Providing background on consolidation rationales and options.
- Discussing hardware and software considerations.
- Creating a decision tree based on key factors.
- Including a sample case study demonstrating a consolidation project.

Rationales for Consolidation

Consolidation projects are typically started to achieve specific goals such as creating room for new servers or reducing operating expenditure. These goals can be broadly grouped into the following categories:

- Lack of space in the data center
- Reducing costs and improving efficiency
- Standardization and centralization
- IT agility
- Green IT

In order to create a consolidation plan, the key deliverables for your project should be decided ahead of time so you have time to make the optimal choices for the execution plan.

Lack of Space

Lack of space in the data center is one of the more direct reasons for needing to consolidate. As companies grow, so too do their hardware needs and with them the need for someplace to store all of that hardware. Acquiring new space through expansion of the data center or the creation of new data centers can result in significant capital expenditure. Consolidation here either targets under-utilized machines or focuses on enabling consolidation by upgrading to newer machines to take advantage of higher performance and scalable growth.

Reducing Cost and Improving Efficiency

Through application consolidation, hardware will run closer to capacity, reducing inefficiencies and allowing for fewer machines. Reducing capital and operating expenditure is one of the biggest factors driving companies to consolidate. Upgrading to fewer machines and newer hardware allows for reductions in rack space, power, and cooling needs. Database sprawl is also minimized, because the machines are more easily centrally managed. Central management provides better control, reducing administrative overhead and maintenance overhead, and it can also help reduce licensing costs.

Standardization and Centralization

One of the issues in database sprawl is an inconsistent approach to database schema creation and database management. Consolidation can be used to pull these various databases into a centrally managed system and also act as a forcing function for standardization, because the applications involved need to coexist on a common platform and thus share common schemas and management infrastructure. By having a common set of requirements and methodologies, administrators are able to take advantage of predictable workflows for patching and configuration, better audit control over security configuration, and streamlined hardware requirements. This also increases the opportunity to make improvements in ease of deployment and provisioning as well as application interoperability.

IT Agility

One additional consideration for consolidation is investing in building a long-term dynamic and power-aware IT infrastructure that allows for better control and flexibility of computing resources in terms of their placement, sizing, and overall utilization. If you move applications onto newer hardware, those applications can take advantage of improved performance and reliability from the new machines, which can be better configured for high availability scenarios, reducing downtime and allowing for rolling upgrades.

Green IT

One of the growing motivations in reducing power and thermal costs is to also focus on providing a “greener” operating environment. This is similar to the motivations for reducing cost and increasing efficiency, but the ultimate goal is environmental benefit rather than cost savings. Consolidation plays an important role here in reducing the data center footprint. Fewer computers and fewer idle machines result in lower power consumption and a reduced need for cooling. New hardware can also provide better energy efficiency as well, because it can take advantage of more power efficient technologies. The Windows Server® operating system also provides capabilities for CPU throttling, and new features in Windows Server 2008 R2 such as core parking further increase efficiency and lower overall energy usage. A study by Microsoft’s IT department found that consolidating onto new servers resulted in reducing power requirements by 3 million volt amps (with an added bonus of also providing a savings of $11 million a year in operating costs). For more information, see the article Green IT in Practice [http://msdn.microsoft.com/en-us/architecture/dd393309.aspx] (http://msdn.microsoft.com/en-us/architecture/dd393309.aspx).

Candidate Applications for Consolidation

The term *application* can refer to a wide variety of services. This paper will focus on consolidation strategies for online transaction processing (OLTP) applications storing data in the SQL Server Database Engine. OLTP applications typically focus on fast response times with smaller but more frequent queries and updates dealing with relatively small amounts of data. An example is an order entry system. For the rest of this paper, the term *application* is generically used to refer to an OLTP application storing data in a SQL Server database, and the consolidation strategy applies primarily to the consolidation of the SQL Server instance supporting this application.

Early in the process of a consolidation project, you will create a profile to help identify which applications are good candidates for consolidation. Then you can identify the applications that fit this profile. Some general traits that make an application a good candidate for consolidation are low machine resource utilization, moderate performance requirements, little active development, and low maintenance costs. Another factor to consider is the impact on the application's network and I/O latency, because both the network and storage resources become shared as part of consolidation.

Note: In most cases, Tier 1 applications with stricter performance (especially around I/O) and high availability requirements are not ideal candidates for consolidation.

New in SQL Server 2008 R2

SQL Server 2008 R2 offers several features that can assist you in your consolidation efforts.

SQL Server Control Point

To help manage SQL Server sprawl, SQL Server 2008 R2 introduces the SQL Server Control Point, which is a single location for managing and deploying SQL Server data-tier applications and enrolling SQL Server instances for centralized views over resource utilization. Both SQL Server instances running on a physical machine or in a VM can be enrolled and viewed. The control point also enables the administrator to apply policies to identify consolidation candidates. For more information about the SQL Server Control Point, see this article [http://msdn.microsoft.com/en-us/library/ee210548(SQL.105).aspx] (http://msdn.microsoft.com/en-us/library/ee210548(SQL.105).aspx) in SQL Server 2008 R2 Books Online.

Data-Tier Application

A data-tier application is a new unit for developing, deploying, and managing databases. Registering a database as a data-tier application allows the database to be managed by the SQL Server Control Point. For more information, including an overview of data-tier applications, see Understanding Data-tier Applications [http://msdn.microsoft.com/en-us/library/ee240739(SQL.105).aspx] (http://msdn.microsoft.com/en-us/library/ee240739(SQL.105).aspx) in SQL Server 2008 R2 Books Online.

Scalability Beyond 64 Logical Processors

SQL Server 2008 R2 introduces the ability to support more than 64 logical processors. This ability allows for more applications to be potentially consolidated onto a single large machine. Note that if you are considering virtualization, the Hyper-V™ virtualization technology has a separate limitation on processor support from Windows Server and SQL Server and currently only supports up to 64 logical processors on the host operating system and up to 4 virtual processors in the virtual machine.

SysPrep Support

SQL Server 2008 R2 provides the ability for the user to install an unconfigured image of SQL Server 2008 R2 in preparation for running the SysPrep utility on a Windows® operating system image. This enables the creation of standardized Windows deployment images with SQL Server preinstalled. IT administrators can use SysPrep and SQL Server 2008 R2 to create a consistent consolidation environment.

Options for SQL Server Consolidation

Each consolidation option provides a certain degree of isolation, which may impact the number of applications that can be consolidated to one machine (referred to as *density*). Typically having higher isolation allows for greater flexibility in leveraging features, but it may increase management cost and reduce the density limit. Achieving higher density results in lower isolation in order to optimize resources and reduce management cost. This can be cost-effective, but it may reduce the ability to leverage certain features and increase the potential for resource contention.

Database

In database-level consolidation, multiple applications share (store data) in one SQL Server instance; each application is contained within its own database or set of databases. Because all of the applications are in the same SQL Server instance, this also means that all applications share the same SQL Server patch level (that is, the major and minor version of SQL Server) and all server-level objects such as **tempdb**. Thus, all applications also share the same service account and, as a result, have the same access level to the host system. This option is attractive in terms of reducing management and licensing costs because fewer SQL Server instances need to be maintained. In SQL Server 2008 R2, databases can be registered as data-tier applications for managing within a SQL Server Control Point after the host SQL Server instance is enrolled as a managed instance for centralized management.

Instance

In instance-level consolidation, multiple applications are moved onto a single physical server with multiple SQL Server instances; each application is contained within its own SQL Server instance. This option provides isolation of the SQL Server instance binaries, allowing for each application to be at different patch levels (major and minor version level). However, potential exists for application conflicts because system resources (mainly CPU, memory, and I/O) are shared, although tools such as the **CPU affinity mask** and **max server memory** settings can help provide resource isolation. Database system administration is isolated, but Windows system administration is shared for the host server. Each SQL Server instance on the machine can be enrolled within a SQL Server Control Point for management.

Virtualization

In this approach, applications are migrated from their physical server into a virtual machine (VM).A single machine hosts multiple VMs, and each VM hosts a single SQL Server instance. Because the VM can act as a dedicated physical machine, this approach provides an easier migration of the source environment to the consolidation environment. VMs are fully isolated from other VMs and communicate with other servers on the network as if they were physical machines. Optimal resource governance between multiple VMs is automatically managed by the hypervisor. Additional high availability options are also available with features like Microsoft Hyper-V Live Migration. While this approach results in fewer physical servers to manage, it does maintain as many operating system and SQL Server images as the source environment. The SQL Server instances within the VM can be enrolled in a SQL Server Control Point for management.

Other Consolidation Options

Other possibilities include further optimizations on an existing approach such as schema-level consolidation or hybrid approaches, such as mixing the instance consolidation and database consolidation approaches or having multiple SQL Server instances in a VM. These approaches may require management at several levels, but they may provide more flexibility with a blend of the advantages and disadvantages of each approach. The key decision factors are similar to the higher-level consolidation options mentioned previously, so this paper will focus only on those.

Hardware and Software Considerations

It may be worthwhile to establish a standardized server type or configuration for the consolidation machine. Standardization can help streamline the ordering process for additional machines, and it presents a common set of maintenance and configuration requirements.

Potential Bottlenecks

Consolidation can introduce bottlenecks for application performance. An unconsolidated application is likely to have a dedicated physical machine with its own CPU, RAM, storage, and network devices and few if any other applications running on top of it; a consolidated application resides on a machine that shares all of these resources with other applications. As a result, it is important to size the consolidation candidates beforehand and choose a consolidation machine that has multiple CPU cores, a large amount of memory, and sufficient storage and network adapters to handle the load. The SQL Server Control Point can be helpful here for viewing historical data on CPU and storage utilization. An in-depth discussion of how to configure the storage and network layers for the server is beyond the scope of this paper, but the paper assumes that the application is not constrained by I/O limitations and that sufficient network bandwidth is available. Please consult with your hardware provider for more information here.

The machines should be selected with room to grow. For example, they should include capacity to add additional CPU cores, memory, and storage, and they should have PCI slots available for additional network and storage controllers. For virtualization, we generally recommend using a fixed size virtual hard disk (VHD) or a pass-through disk because dynamic VHDs can cause additional I/O overhead. For more information, see “Case Study” later in this paper.

New Virtualization Technologies

This paper assumes that the version of Hyper-V included with Windows Server 2008 R2 is used as the virtualization technology. The features discussed here apply specifically to that version. The same principles may apply to other virtualization solutions, but these solutions are not discussed in detail. For more information about Hyper-V, see Virtualization with Hyper-V [http://www.microsoft.com/windowsserver2008/en/us/hyperv-main.aspx] (http://www.microsoft.com/windowsserver2008/en/us/hyperv-main.aspx).

It is important to note that Hyper-V does have some limitations. Hyper-V on the host operating system is only supported on x64 processor architectures and requires hardware-assisted virtualization support (Intel VT or AMD-V) and hardware data execution prevention (DEP, also called Intel XD bit and AMD NX bit). Also, the guest operating system is currently limited to accessing four virtual CPUs. This is unlikely to be a problem for most consolidation candidates, but it is worth considering if the application is expected to substantially grow in usage.

Virtualization technology is, however, constantly improving. One scalability and performance feature of newer processors is second-level address translation (SLAT), also known as nested paging. AMD refers to this technology as NPT, and Intel calls this EPT in their respective processors. Choosing virtualization as a consolidation path does not require SLAT, but the application will certainly perform better.

Figure 1 shows how application workload can benefit from the use of SLAT to achieve linear scale and improved performance.

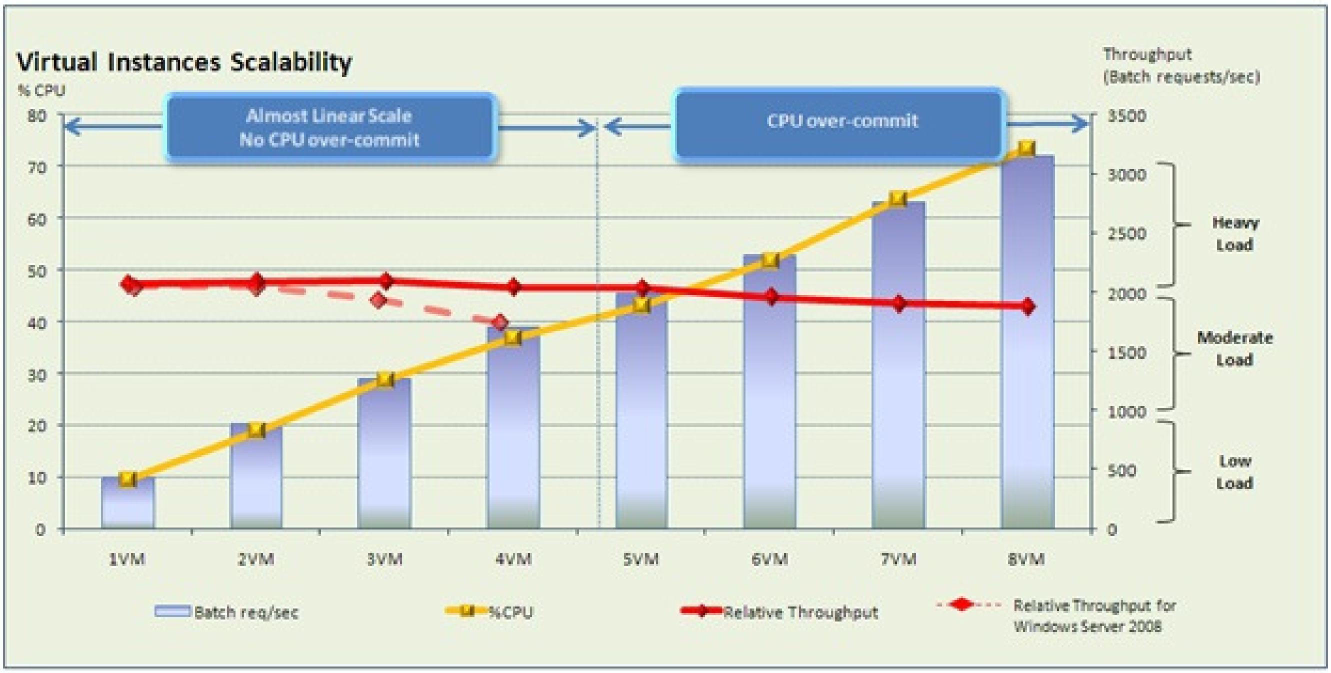

Figure 1: Performance advantage using SLAT processors

These numbers were derived using a 16 core machine, with each VM configured to have 4 virtual processors and 7 gigabytes (GB) of RAM with fixed size VHD for storage. The host operating system was Windows Server 2008 R2.

For comparison, the same workload was run using the Windows Server 2008 implementation of Hyper-V (shown by the dotted red line), which does not take advantage of SLAT. The graph shows that throughput began to suffer after three virtual machines were added.

Hardware Sizing

When consolidating, you should generally consider having as much memory on the target consolidation server as the application was using on the original server. Note that the actual minimum memory required for the application may be less than the total available server memory if the application was unbounded; you may need to perform some analysis to find the minimum amount needed to run the application without affecting performance. Thus if four applications that previously used up to 2 GB of RAM each are consolidated onto a single server, the new server should have at least 8 GB of RAM available. A similar principle applies to processors.

Newer processors may reduce the need for the application to use as many processors as it previously had and many applications under-utilize CPU processing power anyway. Take this under-utilization into account when you plan hardware needs for your consolidation project. To begin, look at all applications that significantly under-utilize CPU, pick the one that utilizes the most processors, and take that number of processors as a base. Add to this the processors being used by the applications with higher-utilized CPUs. The consolidation server should have at least as many processors available. You can perform further analysis of CPU utilization to refine your estimates. As mentioned previously, you should always leave room for peak performance or application usage growth. Targeting approximately 50 percent utilization is a good starting point. The SQL Server Control Point can be helpful in collecting and viewing this data. Your hardware vendor may also have sizing tools to help you choose the appropriate consolidation server.

How to Choose a Consolidation Strategy

Choose your strategy based on your organization's priorities and how different consolidation options support those priorities. The key considerations for choosing a consolidation strategy can be broadly divided into the following categories:

- Security
- High availability and disaster recovery
- Resource management
- Density
- Manageability

How important each of these factors is depends on the priorities for your organization's consolidation effort.

Security

Security is a critical factor in creating a consolidation plan. Regulatory requirements around disclosure of information, compliance, separation of responsibilities, and privacy are important to recognize and establish up front, because these policies are usually set externally and monitored by auditors. Improper migration of applications can cause problems that are expensive and difficult to rectify. As a result, it is very important to identify what information is stored and who needs to be notified if that information is accessed improperly or lost so that the appropriate consolidation strategy can be decided. If an application has very strict security requirements, it is an ideal candidate for a virtualized approach to consolidation because the virtual machine has almost the same security isolation options as if the application had a dedicated physical host. Table 1 shows how security requirements are handled in different consolidation options.

Requirement	Virtualization	Instance	Database
Equivalent to having a dedicated physical machine	Yes	No	No
Isolation of local Windows accounts	Yes	No	No
Isolation of SQL Server logins	Yes	Yes	No
Isolation of SQL Server binaries	Yes	Yes	No
Data protection through Windows BitLocker® drive encryption	Yes	Partial – no isolation between applications	Partial – no isolation between applications
Data protection through Windows Encrypting File System	Yes	Yes – if instances have separate service accounts	Partial – no isolation between applications
Data protection through Microsoft SQL Server Transparent Data Encryption	Yes	Yes	Partial – all root certificates are stored in master
Data protection through Windows permissions	Yes	Yes	Partial – SQL Server service account and files shared for host instance
Data protection through SQL Server granular encryption	Yes	Yes	Yes
Data protection through SQL Server granular permissions	Yes	Yes	Yes
Auditing of actions with SQL Server Audit	Yes	Yes	Yes

Table 1: Comparison of security considerations across consolidation options

In general, it is better to keep applications with different security requirements separate. For example, an application with customer data that requires restricted access should not be consolidated onto a machine that hosts an application that is regularly accessed by users who do not normally have access to the customer data. Such consolidation increases the risk that an improperly configured login or permission would grant access to the private data. This is even more important with the database-level approach; because the Windows accounts and SQL Server logins are shared and the binaries themselves are the same, a security exploit in one database can potentially affect another.

Instance-level consolidation provides an additional layer of protection, because the binaries and the SQL Server logins are separate, but the instances still share the same Windows accounts and operating system configuration. At the instance level, we recommend that you use different service accounts for each instance to reduce security risks from one process affecting another (SQL Server 2008 and SQL Server 2008 R2 take advantage of the service security identifiers (SID) support provided by Windows Server 2008 and Windows Server 2008 R2 to mitigate these risks).

Before you decide on an approach, it is important to first identify where the specific vulnerabilities exist for the application. Certain types of vulnerabilities can help you rule out some consolidation approaches. For example, hard-coded dependencies on the SQL Server system administrator account, other server roles, credentials, or any other server objects (for example, **tempdb** or **msdb**) need to be explicitly identified, because these applications have an increased risk of inadvertent information disclosure. These applications need to be either modified or consolidated with an approach other than the database approach. Even if there isn't sensitive information, these dependencies increase the risk of cross-application corruption or overwriting of data.

If sensitive information is being stored, it is important to identify what is being used to protect this data. It might be operating-system-based mechanisms such as Windows BitLocker and Windows Encrypting File System (EFS), or it might be database mechanisms such as SQL Server Transparent Data Encryption (TDE). If the application relies on operating-system-based mechanisms, database-level or even instance-level consolidation options may not be viable, because these share the same operating system environment.

High Availability and Disaster Recovery

As part of creating the consolidation plan, consider each application's high availability and disaster recovery requirements. The virtualization approach leveraging Hyper-V has an advantage in minimizing planned downtime through Live Migration because the application can remain active during planned failover. Other high-availability solutions may require applications to restart or clients to reconnect after failover. For more information about Live Migration, see the Hyper-V Live Migration Overview and Architecture [http://www.microsoft.com/downloads/details.aspx?FamilyID=fdd083c6-3fc7-470b-8569-7e6a19fb0fdf&displaylang=en] white paper (http://www.microsoft.com/downloads/details.aspx?FamilyID=fdd083c6-3fc7-470b-8569-7e6a19fb0fdf&displaylang=en). Live Migration also requires that the hosts share processors from the same manufacturer. See the Live Migration white paper for details.

All three approaches can leverage the various high-availability features built into SQL Server such as failover clustering, database mirroring, and replication. The unit of failover is different depending on the feature used. Table 2 compares these features across consolidation options.

Feature	Virtualization	Instance	Database
Application remains available during planned host machine downtime without application restart	Yes – via Live Migration (database mirroring can also be used)	Yes – via database mirroring	Yes – via database mirroring
Application remains available during planned host machine downtime without client reconnect	Yes – via Live Migration	No	No
Application can be migrated between machines without downtime (restart or reconnect)	Yes – via Live Migration	No	No
SQL Server failover clustering	Yes	Yes	Partial – failover is at the instance level
SQL Server log shipping	Yes	Yes	Yes

SQL Server database mirroring	Yes	Yes	Yes
SQL Server replication	Yes	Yes	Yes

Table 2: Comparison of high-availability features across consolidation options

You can place applications that require similar levels of availability on the same machine. Such grouping can take advantage of your best hardware, and it can help focus management resources on maintaining those applications. However, you must remember that the high availability technology determines the level at which failover takes place, and consider your choice of consolidation strategy accordingly. Virtualization or dedicated hardware may be the best choices in this scenario. For example, if SQL Server failover clustering is the high availability solution, database-level consolidation may not be the best choice, because failover will occur at the instance level. If you have applications that are consolidated at the database level, these applications will need to rely on health monitoring based on the entire instance failing over. Conversely, instance-level or even virtualization consolidation strategies allow the application to take advantage of high availability features that are delivered at the database level, such as database mirroring. Virtualization, for example, allows the application to take full advantage of Live Migration, SQL Server failover clustering, database mirroring, and other high availability features simultaneously to control specific degrees and stages of failover and availability. Finally, because the applications all share one machine, a failure in one application could cause machine issues, resulting in downtime for all of the other applications as well. Thus it is potentially better to rely on virtualization or dedicated hardware to achieve a high degree of isolation and avoid issues where failures from other applications affect availability.

Resource Management

The consolidated server should be able to handle one or more applications at peak usage and applications that suddenly require more resources than normal, and it should be able to prevent situations where the resource contention caused by resource usage spikes could impact the reliability and consistency of other applications residing on the server. The consolidated server should also be able to handle any sustained usage growth from the application as well. Several features are available to handle resource management.

Virtualization provides perhaps the most specific boundaries, because CPU and memory allocation must be specified for the entire VM container. Both of these settings can be modified later, but you may need to take the VM offline to perform these changes. The benefit of this approach is that the resources are contained and isolated within the VM (with the exception of the CPU, which can be over-committed), which thus reduces the impact of one application experiencing nonaverage workloads affecting other applications on the server. On the downside, these resources will be allocated to the VM regardless of whether or not they are fully utilized. In addition, the guest operating system of the VM itself will consume some overhead of the allocated resources, and the host operating system will also require an additional allocation of resources although these are generally relatively small (please refer to the minimum operating requirements). One virtual processor is recommended to be mapped to one physical processor. It is fairly safe to over-commit processors (that is, to have multiple virtual processors map to one physical processor) but performance should be monitored. You should not over-commit memory, because doing so can create bottlenecks which impact performance. In fact, you cannot over-commit memory in Hyper-V.

Instance-level and database-level consolidation options provide direct access to the consolidated server's physical hardware, which may help scalability by providing support for hot-add CPU and memory (adding a logical processor or adding system memory on a live, running server), for example. However, direct access also creates an opportunity for resource contention. To address this, SQL Server provides the **max server memory** and **CPU affinity mask** settings to set limits on how much memory and how many logical processors the SQL Server instance can use. For more information about setting **max server memory**, see Server Memory Options [http://msdn.microsoft.com/en-us/library/ms178067.aspx] (http://msdn.microsoft.com/en-us/library/ms178067.aspx) in SQL Server 2008 Books Online. For more information about setting **CPU affinity mask**, see affinity mask Option [http://msdn.microsoft.com/en-us/library/ms187104.aspx] (http://msdn.microsoft.com/en-us/library/ms187104.aspx) in SQL Server 2008 Books Online.

Note: **CPU affinity mask** is ignored in a VM.

The other consideration for resource contention is the interaction between the application and **tempdb**. If multiple applications are consolidated as databases and these have dependencies on **tempdb**, I/O bottlenecks on **tempdb** can cause performance issues. If this is the case, consider using either instance or virtualization consolidation or modifying the applications. For more information about **tempdb**, see tempdb Database [http://msdn.microsoft.com/en-us/library/ms190768.aspx] (http://msdn.microsoft.com/en-us/library/ms190768.aspx) in SQL Server 2008 Books Online.

If no contention exists between applications for server-level objects, database-level consolidation can be easier to manage from a resource standpoint because the administrator only needs to configure a single SQL Server instance per machine. This instance is thus able to use the full resources of the machine without concern for sharing CPU or memory for other SQL Server instances. The applications within the instance may still contend for resources, but the Resource Governor feature introduced in SQL Server

2008 can be used to manage workloads (groups of queries) through limits on CPU and memory resource consumption and prioritization between workloads. For more information about Resource Governor, see Managing SQL Server Workloads with Resource Governor [http://msdn.microsoft.com/en-us/library/bb933866.aspx] (http://msdn.microsoft.com/en-us/library/bb933866.aspx) in SQL Server 2008 Books Online. Resource Governor can also be used in the other two consolidation options to further tune performance within the SQL Server instance, but it cannot be used across SQL Server instances or VMs.

Consideration	Virtualization	Instance	Database
Isolation of **tempdb**	Yes	Yes	No
Isolation of server level objects (credentials, linked servers, **msdb**, SQL Server Agent jobs, and so on)	Yes	Yes	No
Hard limits on CPU and memory usage set per application	Yes	Yes	No
Use of Resource Governor to provide query prioritization within a SQL Server instance	Yes	Yes	Yes
Hot-add CPU	No	Yes	Yes
Hot-add memory	No	Yes	Yes
Hot-add storage	Yes	Yes	Yes

Table 3: Comparison of resource isolation considerations across consolidation options

Density

In the context of consolidation, density is the number of applications that can be consolidated to a single machine. VMs, SQL Server instances, and SQL Server databases all have different degrees of overhead, which affects consolidation density. VMs have the highest overhead, because a full operating system is maintained for each application. At the instance level, operating system resources are shared, but each application has an independent instance running, which is its own independent service. Database-level consolidation provides the lowest overhead, because all other resources are shared with the other databases on the single instance. Note also that SQL Server is currently limited to a maximum of 50 instances per operating system environment (physical or virtual). Hyper-V has a limit of 64 VMs per node and a SQL Server instance has a limit of 32,767 databases per instance.

Two data points to capture when measuring density are throughput and response time. These points determine the *density limit*, which this paper defines as the moment at which adding an additional application causes the average response time or the average throughput for one or more other applications to become significantly lower than it was with the original hardware.

Ensure that the target consolidation server includes extra capacity. You should not assume that the server will run at 100 percent average CPU capacity after consolidation; it still needs to handle peak workloads, increases in users, and increases in operational workloads. The target server should have room for multiple applications to reach their peak workload simultaneously as well as handle any growth that may occur from application usage over time. Keeping 50 percent CPU utilization is generally sufficient, and it provides room for both peak workloads and growth in application usage over time.

The case study described later in this paper compares density across the consolidation options with sample hardware and a simulated application. Table 4 summarizes this case study. The numbers in this table were created by using older hardware to generate an application baseline. The baseline is meant to approximate a production application that is a candidate for consolidation. The baseline was then migrated to the new server, and additional copies of the application were added until either throughput or response time was affected. For more information about how this experiment was conducted, see “Case Study” later in this paper.

Baseline is calculated at 100%. For throughput, exceeding 100% indicates that the application was able to achieve higher average throughput (ability to handle more queries) so higher numbers indicate better results. For response time, a value less than 100% indicates that the client was able to receive a response from the server faster than the client received a response from the baseline server. This number is a percentage of the overall time the baseline server took, so a lower number indicates that the application had a faster overall response time and less latency.

Consolidation method	Number of applications	Throughput	Response time	Host system CPU utilization

Baseline (old hardware)	1	100%	100%	6%
Virtualization	24	+0.8%	80%	24%
Instance	24	+0.6%	58%	20%
Database	24	+0.9%	53%	16%
Virtualization	40	+0.6%	95%	45%
Instance	40	+1.1%	73%	37%
Database	40	+1.3%	55%	34%

Table 4: Density results based on throughput (higher is better) and response time (lower is better) across options

Manageability

Virtual machines provide significant manageability flexibility, because they have all the options of a dedicated physical machine combined with simplicity of management offered by Microsoft System Center Virtual Machine Manager (VMM) and Microsoft SQL Server Control Point. VMM also has a "physical to virtual" utility referred to as P2V. This utility takes a physical machine, converts it into a VM, and then deploys it onto a Hyper-V server. This provides for very low cost of migration, because the entire process is handled automatically. For more information about how to use P2V, see P2V: Converting Physical Machines to Virtual Machines in VMM [http://technet.microsoft.com/en-us/library/cc764232.aspx] (http://technet.microsoft.com/en-us/library/cc764232.aspx) in the Microsoft System Center documentation. For an overview of virtualization with Hyper-V and details about the specific benefits of virtualization manageability, see Virtualization with Hyper-V [http://www.microsoft.com/windowsserver2008/en/us/hyperv-overview.aspx] (http://www.microsoft.com/windowsserver2008/en/us/hyperv-overview.aspx) on the Windows Server 2008 R2 Web site. Some of the specific features that can be leveraged with virtualization are the ability to clone and deploy an application very easily and the use of Live Migration to rapidly deploy applications between machines for dynamic load balancing with zero downtime.

SQL Server 2008 R2 offers new technologies to assist you with consolidation. As mentioned previously, all three approaches can take advantage of the SQL Server Control Point in SQL Server 2008 R2 to provide centralized resource utilization views and policies over managed instances. Another new feature is the ability to convert an existing application into a data-tier application definition; this provides a convenient way to migrate an application's schema and logins because the data-tier application definition is designed to be more portable than a full SQL Server instance, and it encapsulates server-scoped objects such as logins. Note however that this process does not migrate data for the application between servers. You must perform data migration separately using backup and restore or another method. For more information about how to convert an existing database to a data-tier application definition, see How to: Extract a DAC [http://msdn.microsoft.com/en-us/library/ee210526(SQL.105).aspx] (http://msdn.microsoft.com/en-us/library/ee210526(SQL.105).aspx) in SQL Server 2008 R2 Books Online. The data-tier application can then be registered in a SQL Server Control Point for centralized management. Table 5 provides a comparison of some manageability highlights across the consolidation options.

Feature	Virtualization	Instance	Database
Create predefined images	Yes	No	No
"One click" clone environments between development, test, and production	Yes – with SCVMM	No	Partial – can clone data-tier applications
Low cost migration	Yes – P2V utility	No	Partial – depends on how well contained the application is within a database
Dynamic redeployment of	Yes – with	No	No

application without downtime	Live Migration		
Can be managed by the SQL Server Control Point	Yes	Yes	Yes – if registered as a data-tier application
Requires installing SQL Server multiple times	No – can use P2V or cloning	Yes	No
Reduces number of physical servers to maintain	Yes	Yes	Yes
Reduces number of Windows installations to maintain	No	Yes	Yes
Reduces number of SQL Server instances to maintain	No	No	Yes

Table 5: Comparison of manageability features across consolidation options

A Note on Performance

In theory, performance can be a concern when you are choosing a consolidation strategy, given the different operating overheads of the various approaches. In practice, however, performance issues are fairly easy to mitigate as long as the proper analysis is done and the appropriate hardware is selected. Considerations for performance are also included as part of other metrics such as density and resource management, because all applications should at least perform no worse than they did before consolidation. If an application is performing poorly after migration to the target consolidation server, any number of performance tools can be used to analyze and tweak the application. In the worst case, the application can be migrated to a less overloaded server, or additional resources can be added for whatever the application is short of (I/O, CPU, memory, and so on). If the application has a strict service-level agreement (SLA) and requires a specific performance threshold, it may not be the ideal candidate for consolidation, although newer hardware may help maintain performance parity even with consolidation.

Decision Tree

One way to consider the above factors is to build a decision tree. Figure 2 shows a decision tree that is based on the relative features and options that are available for each approach. It steps through each major decision point described earlier.

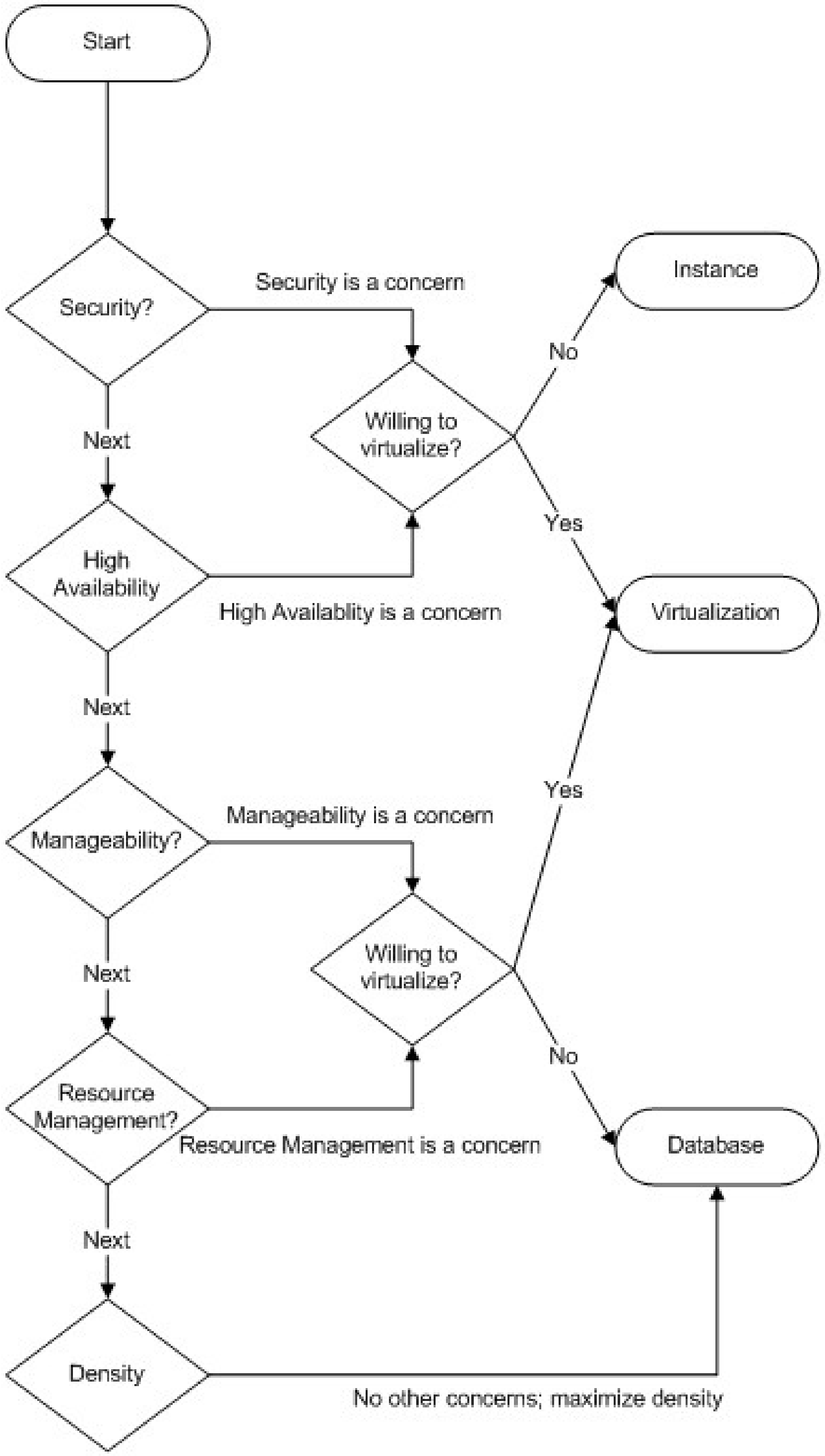

Figure 2: High-level overview of decision tree

This tree captures a high-level positioning of the consolidation options relative to the decision points based on the feature differentiators. Each node is further expanded into a more detailed subtree in Figures 3 through 6. The security portion is shown in Figure 3, high availability in Figure 4, manageability in Figure 5, and resource management in Figure 6. If your organization is optimizing purely on density, consider database-level consolidation; in our case study, the raw numbers indicate that database-level consolidation provides the highest density.

Figure 7 provides a decision tree for determining whether an application can be virtualized. This is a decision point that is often reached as part of the other trees. Note that this is different from the question “Willing to virtualize?” because Figure 7 focuses on identifying technical boundaries for virtualization. “Willing to virtualize?” should be answered based on specific value propositions identified in the subtrees and specific administrative policies your organization may have.

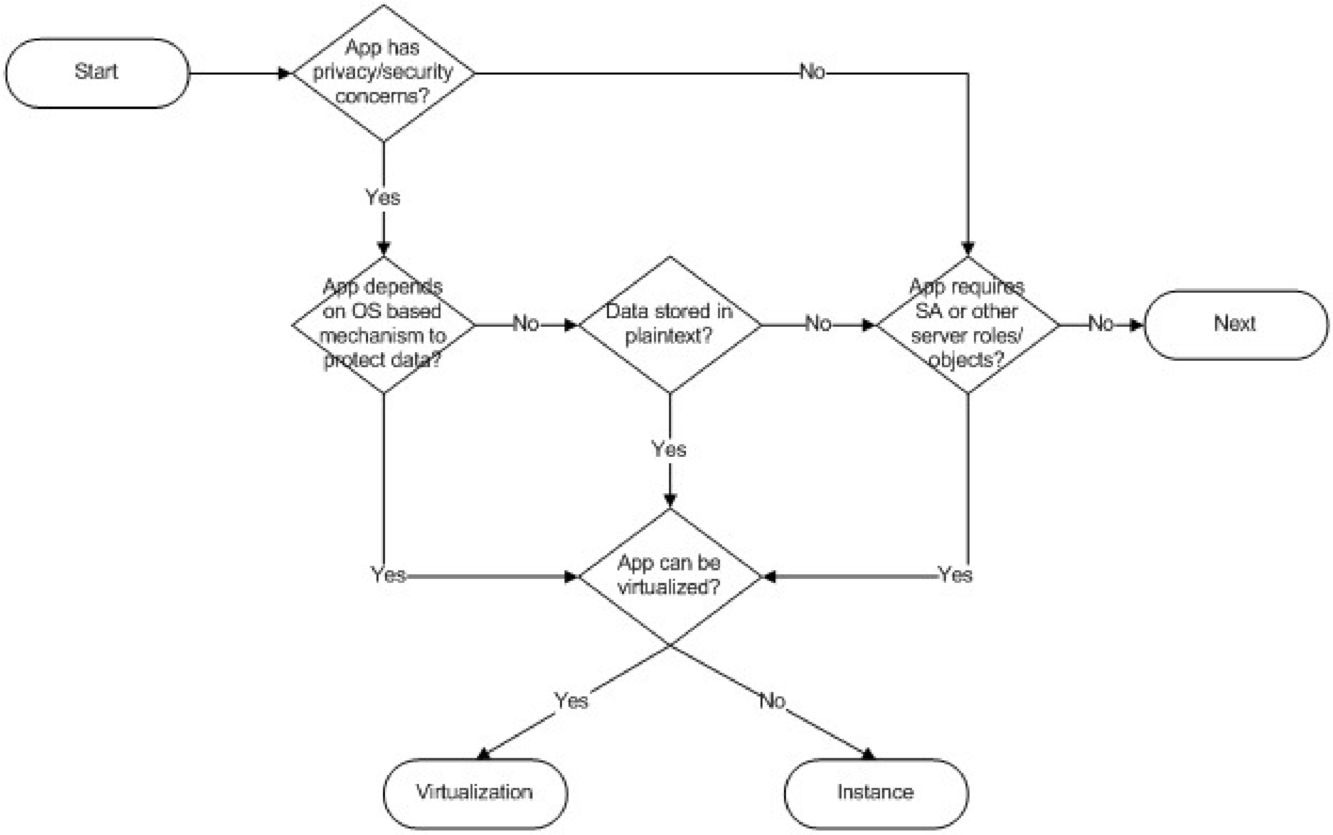
Start
App has privacy/security concerns?
No
Yes
App depends on OS based mechanism to protect data?
No
Data stored in plaintext?
No
App requires SA or other server roles/ objects?
No
Next
Yes
Yes
App can be virtualized?
Yes
Yes
No
Virtualization
Instance

Figure 3: Portion of the decision tree focusing on security

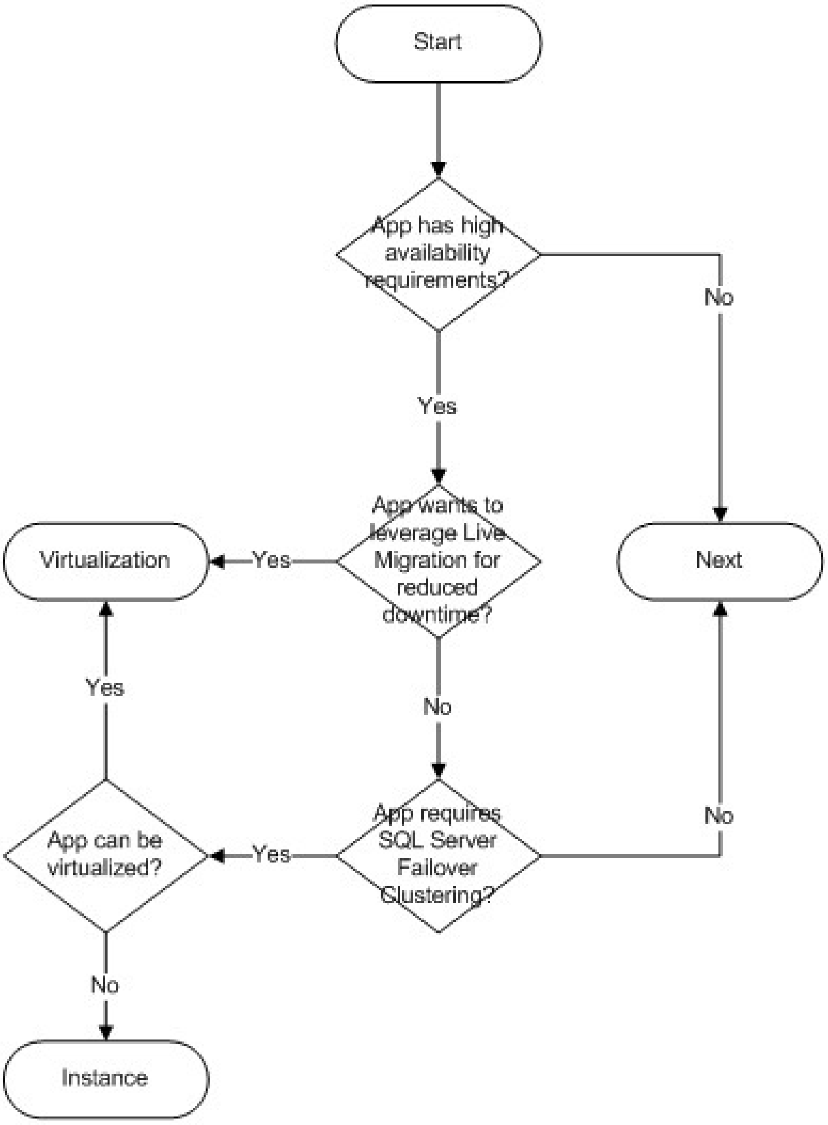
Start
App has high availability requirements?
No
Yes
App wants to leverage Live Migration for reduced downtime?
Virtualization
Yes
Next
No
Yes
App can be virtualized?
Yes
App requires SQL Server Failover Clustering?
No
No
Instance

Figure 4: Portion of the decision tree focusing on high availability

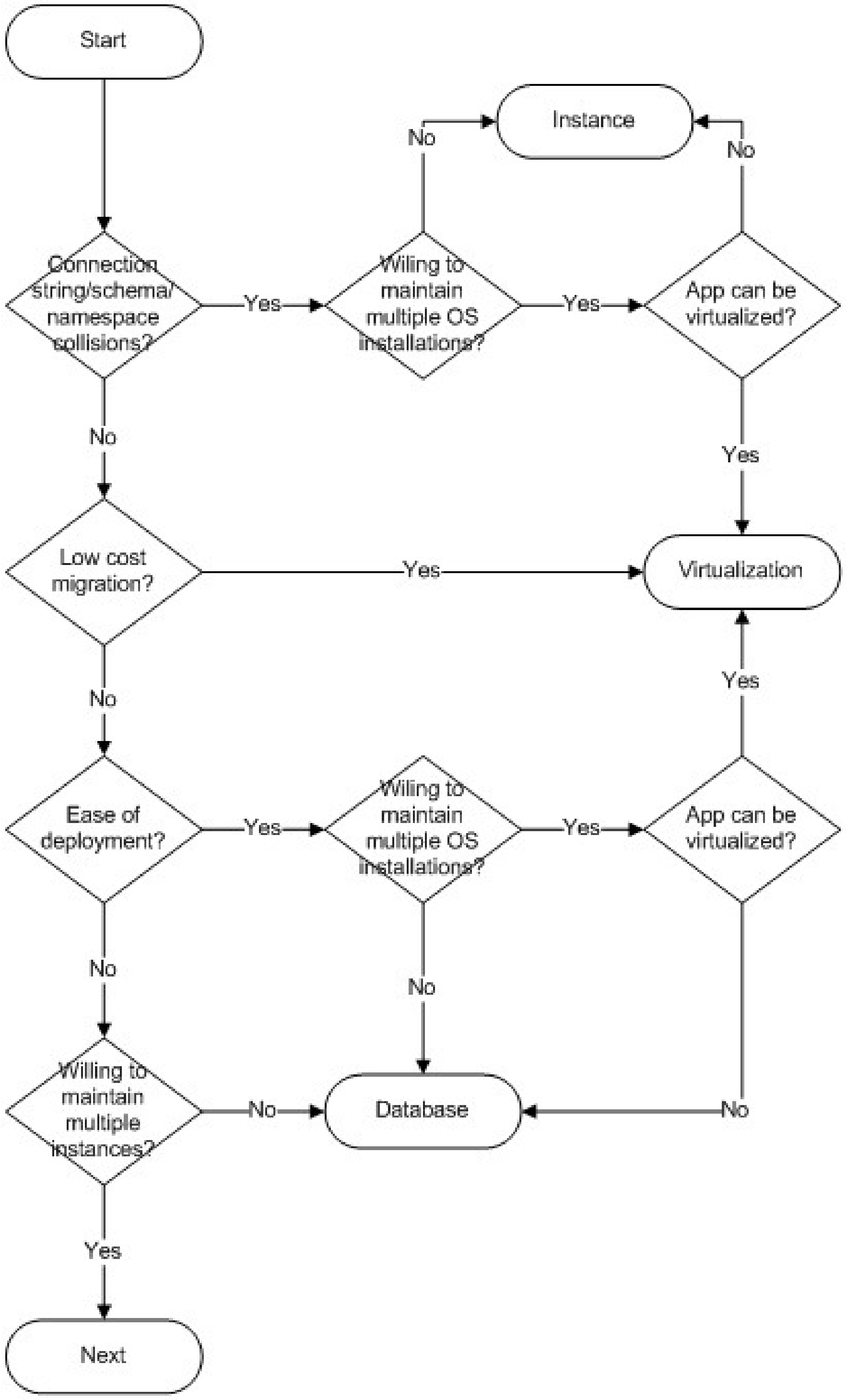
Start
Connection string/schema/ namespace collisions?
Yes
Wiling to maintain multiple OS installations?
Yes
App can be virtualized?
No
Instance
No
No
Yes
Low cost migration?
Yes
Virtualization
No
Yes
Ease of deployment?
Yes
Wiling to maintain multiple OS installations?
Yes
App can be virtualized?
No
No
No
Willing to maintain multiple instances?
No
Database
No
Yes
Next

Figure 5: Portion of the decision tree focusing on manageability

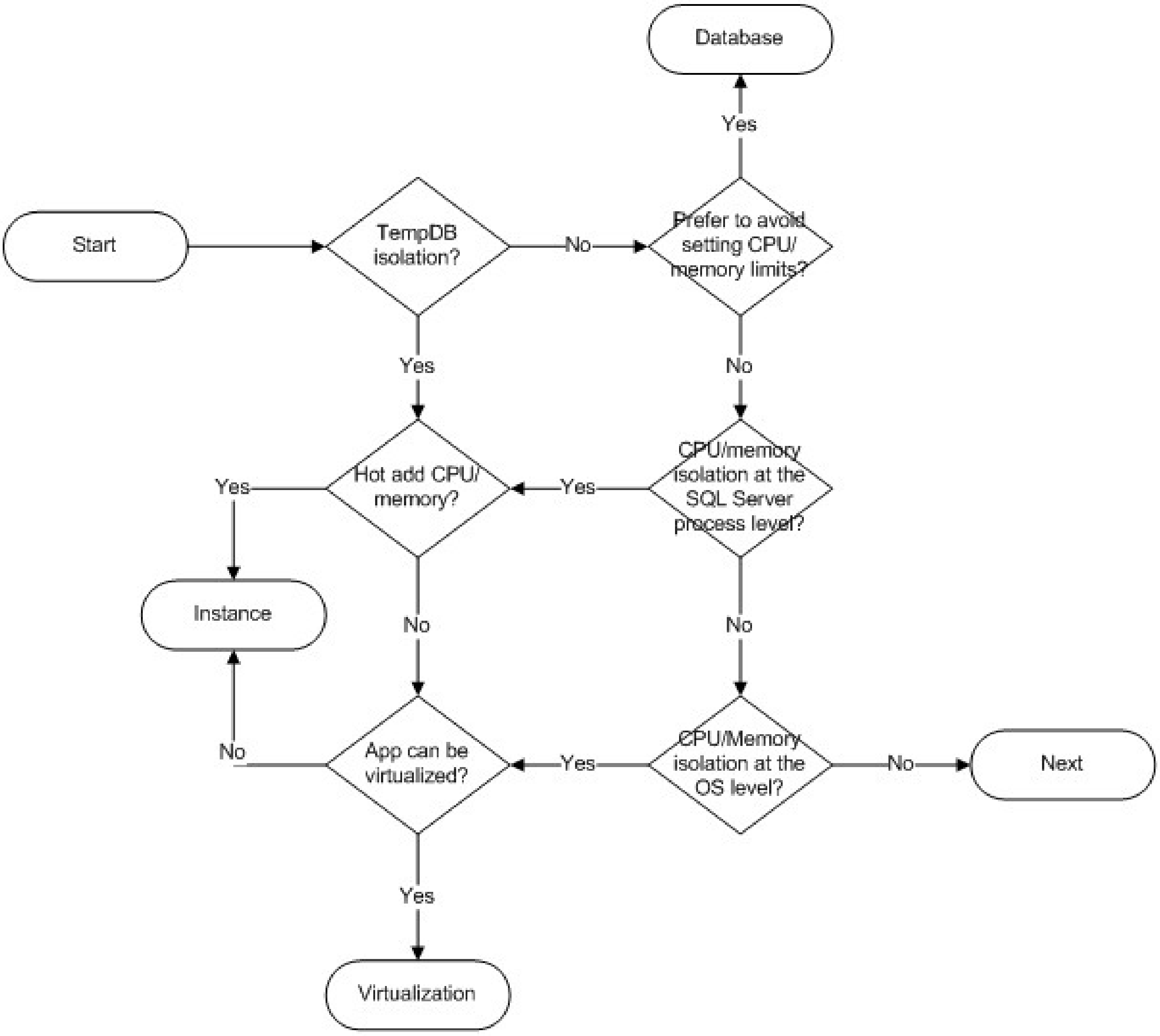
Start
TempDB isolation?
No
Prefer to avoid setting CPU/ memory limits?
Yes
Database
Yes
No
Hot add CPU/ memory?
Yes
CPU/memory isolation at the SQL Server process level?
Yes
Instance
No
No
App can be virtualized?
No
Yes
CPU/Memory isolation at the OS level?
No
Next
Yes
Virtualization

Figure 6: Portion of the decision tree focusing on resource management

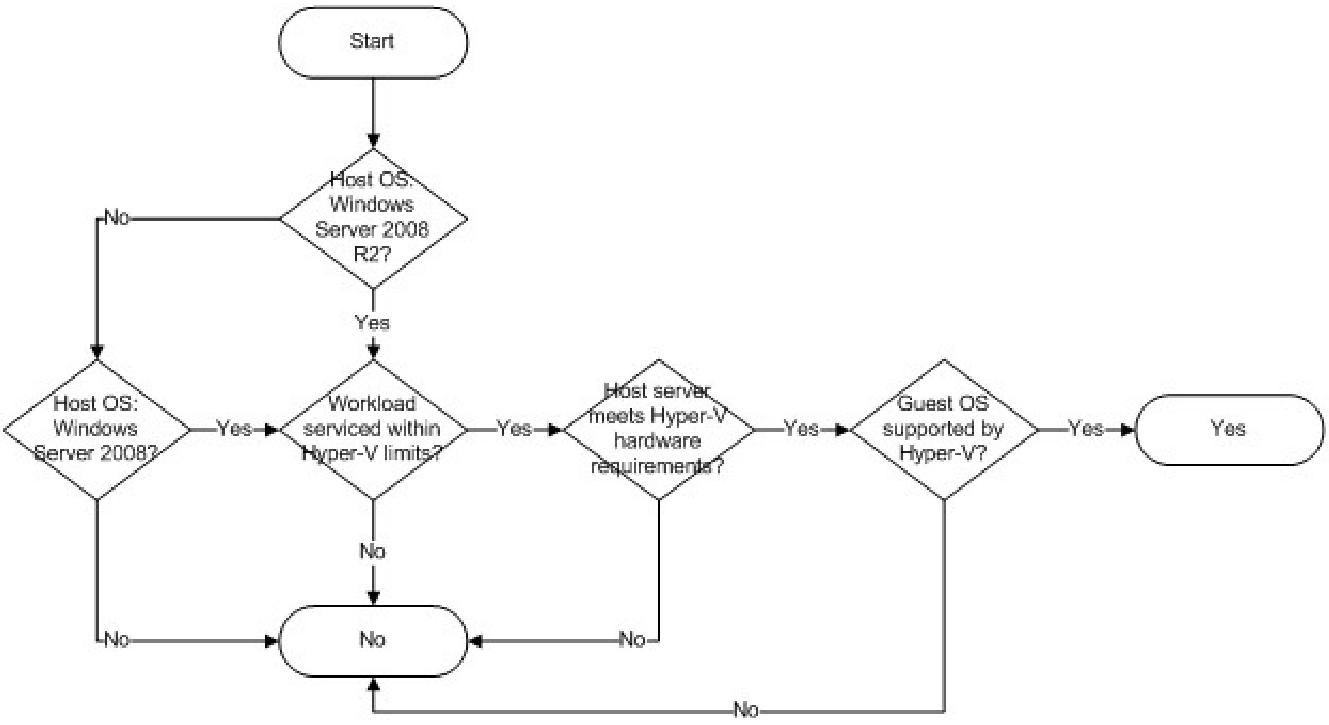
Start
Host OS: Windows Server 2008 R2?
No
Yes
Host OS: Windows Server 2008?
Yes
Workload serviced within Hyper-V limits?
Yes
Host server meets Hyper-V hardware requirements?
Yes
Guest OS supported by Hyper-V?
Yes
Yes
No
No
No
No
No
No

Figure 7: Portion of the decision tree for deciding whether an application can be virtualized

Case Study

Our goal was to conduct an experiment on the different behaviors of each approach with a high consolidation density. We picked an older machine with four logical processors and 8 GB of total system memory available. We targeted a 5-10 percent average utilization rate for this machine. We chose to consolidate this application on to a newer 32-way server with SLAT support, 128 GB of RAM, fifty 135-GB hard drives, and two 1-GBit/s network cards. We then attempted to run as many copies of the application as possible on the new server in an attempt to find out how many applications would be able to run simultaneously and maintain the same performance as on the baseline old machine. The old machine was never stressed, so the workload itself was the limiting factor rather than any resource constraint. As previously noted, the assumption was that the database would have the highest density, because it has the lowest overhead, and that virtualization would have the lowest, because it has the highest overhead.

The first major issues we ran into were storage and network bandwidth. We very easily hit network limits and we constantly had to modify the application to scale back the user workload and avoid saturating the network adapters. The next bottleneck we ran into was storage. Because we were limited to 50 spindles, we had limited options for system, data, and log file partition placement.

For both database and instance consolidation, the applications were configured with separate data and log files. Because we did not have enough individual hard drive spindles to go around, we assigned spindles in a round-robin fashion. All system binaries were located on one partition. CPU affinity was not used, but **max server memory** was used based on the number of applications. We assigned 2.2 GB for each application, which we chose based on the memory usage profile of the application on the original machine. For instance-based consolidation, this means that each instance’s buffer pool was capped at 2.2 GB. For database consolidation, **max server memory** was set to a multiple of 2.2 GB. For example, with 10 applications **max server memory** was set to 22 GB.

For the virtualization consolidation route, each virtual machine was configured with separate fixed size virtual hard disks on different hard drive spindles for data and log. System binaries were also stored on different virtual hard disks on different hard drive spindles from data and log. Each VM only had one virtual processor allocated with 3 GB of memory. At the highest density level, we over-committed processors, because the physical machine only had 32 and we used 40.

One early discovery was the impact of using fixed or pass-through disk for virtualization with a database workload. With the data and log files originally set to dynamic, I/O was significantly affected whenever the disk needed to expand, and we were only able to achieve 75 percent of the consolidation density that we were able to reach using a fixed disk. Using dynamic disks does have an advantage in improving space efficiency, because the size of the VHD only expands as needed, but dynamic disks can have a very large impact on a running database system.

All measurements were done with an industry standard OLTP workload. We considered reaching the density limit as meaning that the throughput and response time for the average application was worse than the original baseline. We found that all three of the consolidation options actually scaled quite well. As shown previously in “Density”, we were able to duplicate this application to a density of 40! We ran out of client machines at this point and thus do not have results for more than 40 applications. As expected, database consolidation did have the lowest overhead and the most room to add additional applications because the CPU utilization and response time was the lowest and the throughput was still the highest at 40 applications. Virtualization had relatively higher response times and used more CPU than the other options, but was not too far off. All three options were able to achieve slightly better throughput than the baseline, and in all cases, CPU utilization never exceeded 50 percent.

The best conclusion to draw is that all three options are viable choices even if an organization is trying to optimize on consolidation density and performance. As a result, it is highly recommended that you make the decision on which option to choose in combination with specific features or requirements from the other factors (security, high availability, manageability, and resource management).

Conclusion

To a certain degree, consolidation is a never-ending project. New machines and new hardware are always arriving, which provide higher consolidation density, longer application availability times, and better performance. Application usage will continue to grow, and new applications will be created to replace or augment the old ones. New factors such as “green IT” policies and practices will also drive consolidation. It is therefore important to create consolidation plans not just for current trends but for future ones as well. By identifying the specific goals that drive consolidation efforts for the organization and making decisions based on the key factors underlying those goals, and keeping in mind the various advantages each consolidation option provides, consolidation can be used not just to achieve short-term objectives, such as reducing costs and creating more space in the data center, but also to create a dynamic and scalable IT infrastructure that makes future consolidation even easier and supports growth for the company.

For more information:

http://www.microsoft.com/sqlserver/ [http://www.microsoft.com/sqlserver/default.aspx] : SQL Server

Web site

http://www.microsoft.com/sqlserver/2008/en/us/server-consolidation.aspx [http://www.microsoft.com/sqlserver/2008/en/us/server-consolidation.aspx] : SQL Server Consolidation site

http://www.microsoft.com/sqlserver/2008/en/us/virtualization.aspx [http://www.microsoft.com/sqlserver/2008/en/us/virtualization.aspx] : SQL Server Virtualization site

http://technet.microsoft.com/en-us/sqlserver/ [http://technet.microsoft.com/en-us/sqlserver/default.aspx] : SQL Server TechCenter

http://msdn.microsoft.com/en-us/sqlserver/ [http://msdn.microsoft.com/en-us/sqlserver/default.aspx] : SQL Server DevCenter

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?

Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

Send feedback [mailto://microsoft.com:25/default.aspx?subject=White%20Paper%20Feedback:%20Consolidation%20Guidance%20for%20SQL%20Server] .