# logo-sql08.gif

Planning, Implementing and Supporting SQL Server Virtualization with Windows Server 2008 R2 Hyper-V and Live Migration

Technical White Paper

**Writers:** Shirmattie Seenarine (Independent Technical Writer) and Ross Mistry (Microsoft Corporation)

**Technical Reviewers:** Sung Hsueh and Prem Mehra

**Editor:** Craig Huber

**Published:** July 2010

**Applies to:** [Microsoft Windows Server](http://www.microsoft.com/windowsserver2008/en/us/default.aspx), [Microsoft SQL Server](http://www.microsoft.com/sqlserver/2008/en/us/R2.aspx), [Microsoft System Center](http://www.microsoft.com/systemcenter/en/us/default.aspx)

**Summary:**

This whitepaper provides prescriptive guidance and detailed steps for planning and implementing SQL Server virtualization with Windows Server 2008 R2 Hyper-V and Live Migration. The document also communicates strategies for managing a SQL Server virtual environment with SQL Server native capabilities and System Center products such as Virtual Machine Manager (VMM) 2008 R2, Configuration Manager 2007 R2, Operations Manager 2007 R2 and Data Protection Manager 2010.

Authors

**Shirmattie Seenarine** is an independent technical writer with more than 10 years of experience. She has contributed to many books, including Introducing SQL Server 2008 R2, [Windows Server 2008 Unleashed](http://www.amazon.com/Windows-Server-2008-Unleashed-Morimoto/dp/0672329301/ref=sr_1_3?ie=UTF8&s=books&qid=1237434991&sr=1-3), [Exchange Server 2007 Unleashed](http://searchsystemschannel.techtarget.com/tip/%20http:/www.amazon.com/Microsoft-Exchange-Server-2007-Unleashed/dp/0672329204/ref=pd_bbs_sr_1?ie=UTF8&s=books&qid=1237789378&sr=8-1), [SharePoint Server 2007 Unleashed](http://searchsystemschannel.techtarget.com/tip/%20http:/www.amazon.com/Microsoft-SharePoint-2007-Unleashed-Michael/dp/0672329476/ref=sr_1_8?ie=UTF8&s=books&qid=1237789378&sr=8-8), and [SQL Server 2008 Management and Administration](http://searchsystemschannel.techtarget.com/tip/%20http:/www.amazon.com/Microsoft-Server-2008-Management-Administration/dp/067233044X/ref=pd_bbs_sr_2?ie=UTF8&s=books&qid=1237789435&sr=8-2). Shirmattie has written many articles, whitepapers, architecture design documents and operational procedures for Fortune 500 organizations including Microsoft, CIBC, Solectron, Network Appliance and Gilead Sciences.

**Ross Mistry** is an Enterprise Architect at the Microsoft Technology Center (MTC) in Silicon Valley. He provides executive briefings, architectural design sessions, and proof of concept workshops to some of Microsoft's largest customers. He specializes in SQL Server and architects solutions focusing on upgrades, migrations, high availability, consolidation and virtualization.

Ross is also an author on many books, articles and whitepapers. His recent books include*: Introducing SQL Server 2008 R2 (MSPress), Windows Server 2008 R2 Unleashed (SAMS), SQL Server 2008 Management and Administration (SAMS), Exchange Server 2010 Unleashed (SAMS) and SharePoint 2007 Unleashed (SAMS).*

Ross is a former SQL Server MVP, well known in the worldwide SQL Server community and frequently speaks at technology conferences and user groups around the world. Conferences include PASS, SQL Connections, Europe PASS, SQL BITS, User Groups and Microsoft. You can follow and contact Ross on Twitter @RossMistry.

Description: Description: http://mtc/marketing/Marketing%20Documents/Video%20and%20Display%20Files/AutoSignature.jpg

Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commit on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2010 Microsoft Corporation. All rights reserved.

Hyper-V, Microsoft, SQL Server, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

[Introduction 6](#_Toc264145091)

[Overview of Technologies and Concepts 7](#_Toc264145092)

[1. Planning SQL Server Virtualization 8](#_Toc264145093)

[1.1. Step 1: Conduct SQL Server Inventory 9](#_Toc264145094)

[1.1.1. Using the MAP Toolkit to Conduct a SQL Server Inventory (Step-by-Step) 10](#_Toc264145095)

[1.1.2. Analyzing the Results 10](#_Toc264145096)

[1.1.3. Generating a SQL Server Summary Report and Proposal 11](#_Toc264145097)

[1.2. Step 2: Understanding Hyper-V Maximum Configurations 13](#_Toc264145098)

[1.2.1. Maximum Configurations for Virtual Machines 13](#_Toc264145099)

[1.2.2. Maximum Configurations for Servers running Hyper-V 13](#_Toc264145100)

[1.2.3. Maximum Configurations for Hyper-V and Failover Clusters 13](#_Toc264145101)

[1.3. Step 3: Determine SQL Server Workloads to Virtualize 14](#_Toc264145102)

[1.4. Step 4 - Determine High Availability, Security and Isolation Strategies 17](#_Toc264145103)

[1.4.1. High Availability Strategies 17](#_Toc264145104)

[1.4.2. Security and Isolation Strategies 19](#_Toc264145105)

[1.5. Step 5 - Determine the Version and Edition of SQL Server and the Operating System 20](#_Toc264145106)

[1.5.1. SQL Server Editions, Components and Features 20](#_Toc264145107)

[1.5.2. Windows Server Editions and Features 20](#_Toc264145108)

[1.6. Step 6: Determine Virtualization Strategy 20](#_Toc264145109)

[1.6.1. Virtualization Strategy: Physical-to-Virtual Conversion 21](#_Toc264145110)

[1.6.2. Virtualization Strategy: Server Consolidation 21](#_Toc264145111)

[1.7. Step 7: Determine Hyper-V Hardware 21](#_Toc264145112)

[1.7.1. Hyper-V R2 Enhancements 22](#_Toc264145113)

[1.7.2. Storage - Pass-through vs. Fixed Size VHD 23](#_Toc264145114)

[1.7.3. Hyper-V R2 System Requirements 23](#_Toc264145115)

[2. Implementing SQL Server Virtualization 24](#_Toc264145116)

[2.1.1. Operating Systems Supported 24](#_Toc264145117)

[2.1.2. P2V Conversion Process 24](#_Toc264145118)

[2.1.3. P2V Conversation Step-by-Step 25](#_Toc264145119)

[2.1.4. Starting and Stopping a SQL Virtual Machine 30](#_Toc264145120)

[3. Implementing Live Migration for SQL Server 2008 R2 30](#_Toc264145121)

[3.1. Enabling Cluster Shared Volumes (CSV) 31](#_Toc264145122)

[3.2. Creating a SQL Server Virtual Machine with Hyper-V 33](#_Toc264145123)

[3.2.1. Creating a SQL Server Virtual Machine with Hyper-V Manager 34](#_Toc264145124)

[3.2.2. Creating a SQL Server Virtual Machine with Virtual Machine Manager R2 37](#_Toc264145125)

[3.3. Initiating Live Migration for a SQL Server VM 39](#_Toc264145126)

[4. Supporting a Virtualized SQL Server Environment 40](#_Toc264145127)

[4.1. Patch and Service Pack Management 41](#_Toc264145128)

[4.1.1. Windows Update: 41](#_Toc264145129)

[4.1.2. Windows Server Update Services 3.0 SP2 (WSUS) : 41](#_Toc264145130)

[4.1.3. System Center Configuration Manager 2007 R2: 42](#_Toc264145131)

[4.1.4. Applying Updates and Service Packs 42](#_Toc264145132)

[4.2. Tools to Manage SQL Server Virtual Machines 43](#_Toc264145133)

[4.2.1. The VMM 2008 R2 Administration Console's Virtual Machine View 43](#_Toc264145134)

[4.2.2. The VMM 2008 R2 Administration Console's Administration View 44](#_Toc264145135)

[4.2.3. Utility Control Point 45](#_Toc264145136)

[4.2.4. Data Collector 46](#_Toc264145137)

[4.3. System Center Operations Manager 2007 R2 Integration 46](#_Toc264145138)

[4.3.1. Viewing Virtual Machine Reports with System Center 47](#_Toc264145139)

[4.4. How to upgrade a SQL Server Instance within a Virtual Machine 48](#_Toc264145140)

[4.5. Backing Up SQL Server Virtual Machines 48](#_Toc264145141)

[4.5.1. Using Data Protection Manager (DPM) 2010 to Protect SQL Server Virtual Machines 49](#_Toc264145142)

5. Conclusion

# Introduction

With the introduction of Windows Server 2008 R2 comes a plethora of innovative capabilities for the Hyper-V platform that will significantly enhance, simplify and streamline your SQL Server virtualization efforts while improving efficiency and minimizing costs. In addition, virtualization can also be used as an enabling factor for SQL Server consolidation. By also leveraging the new capabilities in System Center such as System Center Virtual Machine Manager (VMM) 2008 R2, System Center Operations Manager 2007 R2 and System Center Data Protection Manager 2010, you will establish a more dynamic and flexible virtualized SQL Server infrastructure that will enhance your provisioning, protection and managing experience.

This document is organized into four sections to deliver a straightforward and un-daunting process for planning, implementing and managing your SQL Server virtualization efforts. The document opens with a high level introduction that will acquaint you with the technologies and concepts involved in planning, implementing and managing SQL Server virtualization. The document then expands on these technologies by discussing how to use each capability to simplify your SQL Server virtualization efforts. The four sections include:

* **Part 1 Planning:** This section of the document describes the steps on how to effectively plan a SQL Server virtualization strategy by first communicating how to conduct a discovery of SQL Server computers, instances, and databases in your network. It then covers key considerations that will help you select the most appropriate Hyper-V hardware to support your SQL Server virtualization efforts.
* **Part 2 Implementation:** In this section you will findstep-by-step procedures on how to use the Physical-to-Virtual (P2V) tool to conduct a migration of a SQL Server physical server to a virtual server. Steps on how to create a new SQL Server virtual machine from the ground-up are also described in Part 2.
* **Part 3 High Availability and Live Migration:** Achieving high availability within your SQL Server virtual environment with Live Migration is the focus of Part 3. Step-by-step procedures for implementing Live Migration are covered in this section.
* **Part 4 Ongoing Operations and Support:** The final section of the document discusses several strategies that will help you manage your SQL Server virtual environment. Part 4 also includes an overview of how to use System Center Operations Manager to proactively monitor and protect your SQL Server virtual environment.

# Overview of Technologies and Concepts

This section provides a brief overview of the key technologies and concepts discussed throughout the whitepaper. It also includes the Microsoft homepages for each technology.

* **SQL Server 2008 R2** - SQL Server 2008 R2 delivers several breakthrough capabilities that will enable your organization to scale database operations with confidence, improve IT and developer efficiency, and enable highly scalable and well managed Business Intelligence on a self-service basis for your users. (<http://www.microsoft.com/sqlserver/2008/en/us/r2.aspx>)
* **Windows Server 2008 R2** - Windows Server 2008 R2 builds on the foundation of Windows Server 2008 and delivers valuable new functionality and powerful improvements to the core Windows Server operating system to help organizations of all sizes increase control, availability, and flexibility for their changing business needs. New Web tools, virtualization technologies, scalability enhancements, and management utilities help save time, reduce costs, and provide a solid foundation for your information technology (IT) infrastructure. (<http://www.microsoft.com/windowsserver2008/en/us/default.aspx>)
* **Hyper-V** - With Hyper-V, it's easier than ever to take advantage of the cost savings of   
  virtualization through Windows Server 2008 R2. Optimize your server hardware investments by consolidating multiple server roles as separate virtual machines running on a single physical machine, efficiently run multiple different operating systems in parallel, on a single server, and fully leverage the power of x64 computing. (<http://www.microsoft.com/windowsserver2008/en/us/hyperv-overview.aspx>)
* **Live Migration** - Windows Server 2008 R2 Hyper-V provides greater flexibility with Live Migration. Live Migration is integrated with Windows Server 2008 R2 Hyper-V and Microsoft Hyper-V Server 2008 R2. With Hyper-V Live Migration, you can move running virtual machines (VMs) from one Hyper-V physical host to another, without any disruption or perceived loss of service. IT professionals increasingly look to Live Migration to create a dynamic and flexible IT environment that responds to emerging business needs. Live migration provides the core technology required for dynamic load balancing, VM placement, high availability for virtualized workloads during physical computer maintenance, and reduced data center power consumption. (<http://www.microsoft.com/downloads/details.aspx?FamilyID=FDD083C6-3FC7-470B-8569-7E6A19FB0FDF&amp;displaylang=en&displaylang=en>)
* **System Center Virtual Machine Manager (VMM) R2** - System Center Virtual Machine Manager is a comprehensive heterogeneous management solution for the virtualized data center. Virtual Machine Manager enables increased physical server utilization, centralized management of virtual machine infrastructure, and rapid provisioning of new virtual machines by the administrator and authorized end users. Virtual Machine Manager provides the best solution for leveraging existing IT administrative skills and processes for managing the virtual and physical environment. (<http://www.microsoft.com/systemcenter/virtualmachinemanager/en/us/default.aspx>)
* **System Center Operations Manager 2007 R2** - System Center Operations Manager provides end-to-end service management that is easy to customize and extend for improved service levels across your IT environment. This enables Operations and IT Management teams to identify and resolve issues affecting the health of distributed IT services. End-to-end Service Management is not restricted to Microsoft-based environments. Support for Web Servers for Management (WS-Management), Simple Network Management Protocol (SNMP), and partner solutions allow for non-Microsoft systems and hardware to be included in service monitoring within Operations Manager 2007. (<http://www.microsoft.com/systemcenter/operationsmanager/en/us/default.aspx>)
* **Microsoft System Center Data Protection Manager 2010** - (DPM) provides continuous data protection for virtual machines hosted on servers running Microsoft’s Hyper-V. This protection includes online backup of supported guest virtual machines hosted on clustered or standalone systems; protection of virtual machines during the Live Migration process; and, item level recovery from host-level backup. DPM 2010 offers disk-to-disk, disk-to-tape, and disk-to-disk-to-tape technologies; all of which maintain the business value of a virtualized infrastructure by ensuring that it is better protected and always available. (<http://www.microsoft.com/systemcenter/en/us/data-protection-manager.aspx>)

# 1. Planning SQL Server Virtualization

The planning section will guide you through a clear and concise path of tasks that will equip you with the information necessary to make critical decisions about your virtualized environment. The following steps should be used to successfully plan your SQL Server virtualization strategy:

* Step 1: Conduct SQL Server Inventory
* Step 2: Understand Hyper-V Maximum Configurations
* Step 3: Determine SQL Server Workloads to Virtualize
* Step 4: Determine High Availability, Security, and Isolation Requirements
* Step 5: Determine the Version and Edition of SQL Server and Operating System to Use
* Step 6: Determine the Virtualization Migration Strategy
* Step 7: Determine what Hyper-V Hardware to Utilize

Figure 1 provides a flowchart of the steps involved in planning a SQL Server virtualization project. Complete steps 1 through 7.



Figure 1. SQL Server Virtualization Planning Decision Flow

## 1.1. Step 1: Conduct SQL Server Inventory

Collecting information about your existing SQL Server landscape will help you gain a comprehensive understanding of your SQL Server computers, instances and databases. Next, the evaluation of this information is critical in helping you determine the most appropriate SQL Server virtualization strategy as it exposes not only opportunities, but also constraints and restrictions. Each SQL Server computer, instance and database deployed in your environment should be part of the inventory. The results of the inventory should be comprehensive, including such information as the name of the instance, the name of the computer, the SQL Server components installed, the SQL Server version and edition, and the underlying operating system.

The use of the Microsoft Assessment and Planning (MAP) Toolkit, a free Solution Accelerator, provides a powerful foundation for conducting inventories and assessments, and includes intuitive reports which can expedite the SQL Server virtualization planning process. The MAP tool securely runs within your network infrastructure without requiring the installation of agent software on any SQL Server computer. Once the inventory is completed, the MAP toolkit generates a series of reports and graphs summarizing the results of the assessment. The MAP Solution Accelerator Toolkit can be downloaded from Microsoft's Technet website (link below) and is straightforward to install and use.

### 1.1.1. Using the MAP Toolkit to Conduct a SQL Server Inventory (Step-by-Step)

Conduct a discovery and assessment on computers that have SQL Server installed by performing the following steps:

1. Download the latest Microsoft Assessment and Planning Toolkit from the following link: (<http://technet.microsoft.com/en-us/library/bb977556.aspx>).
2. Install the Microsoft Assessment and Planning Toolkit.
3. Launch the toolkit by selecting Start, All Programs, and then Microsoft Assessment and Planning Toolkit.
4. Data gathered in the inventory and assessments are stored in a SQL Server database; therefore, create a new inventory database or use an existing database. Next, click Ok.
5. Launch the Inventory and Assessment Wizard to discover SQL Server instances including all of the SQL Server components in the environment. The Inventory and Assessment Wizard is invoked.
6. On the Computer Discovery Methods screen, specify one of the five applicable methods for discovering SQL Server instances, and then click Next. For this example, the Scan an IP Address Range option was selected. However, the discovery options include:

* Active Directory Domain Services (AD DS)
* Windows networking protocols
* Import computer names from a file
* Scan an IP address range
* Manually enter computer names

1. On the Scan an IP Address Range screen, specify the IP address ranges of the SQL Server computers you plan on discovering, and then click Next.
2. Regardless of the methods chosen to discover your computers, the wizard uses Windows Management Instrumentation (WMI) to connect to each computer and collect hardware and software information. As a result, any computer in the network included in the inventory and assessment process must have an account that is a member of the local Administrators group on its computer. To add an inventory account to a computer, click New Account, specify the credentials, and then click Next to continue.
3. Review your selections on the Summary screen and click Finish. A Status page will then display and the inventory and assessment process will commence automatically.

### 1.1.2. Analyzing the Results

The Microsoft SQL Server Discovery node located in the Discovery and Readiness tree of the Microsoft Assessment and Planning Toolkit contains a summary of your SQL Server network results as illustrated in Figure 2. The results are organized into four sections. Each section is accompanied by a pie graph and charts that provide for easy representation and interpretation of the results. Section one identifies each version of SQL Server in the network along with the number of computers and instances associated with it. Similarly, section two provides a summary of the different editions in the network. The summary includes the SQL Server Version, SQL Server Edition, the number of computers in your network currently running the version and edition, and the number of instances for each edition.

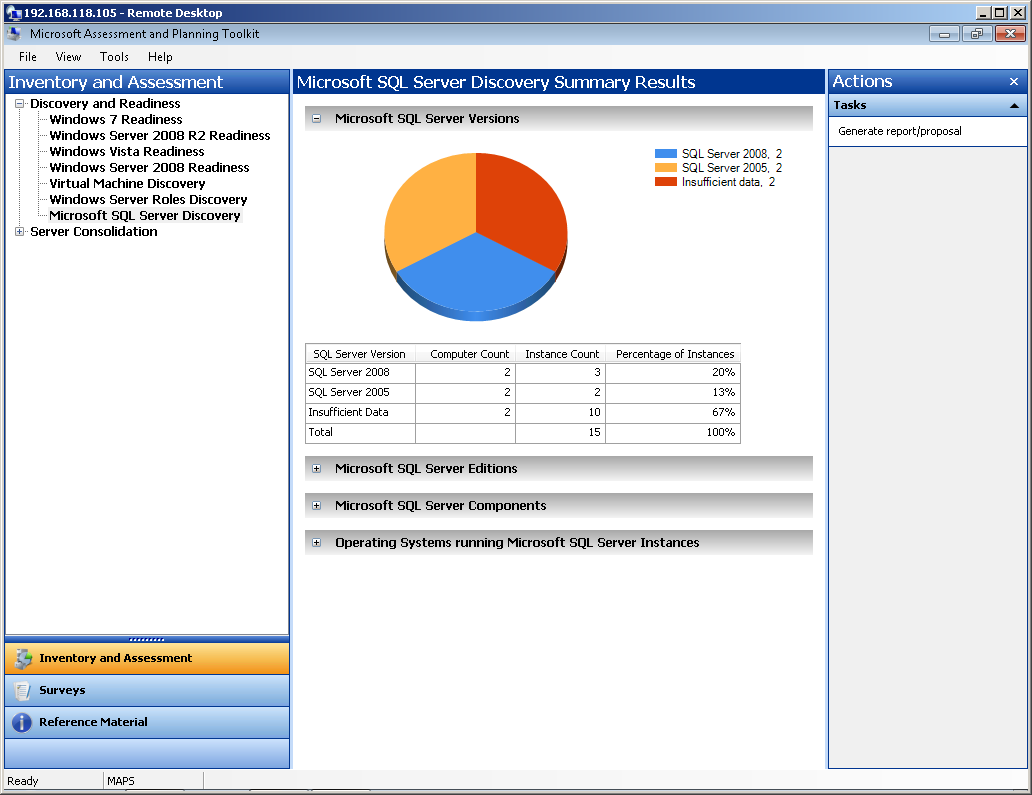


Figure 2. The Microsoft SQL Server Discovery Summary Results Report

In section three, the name of each component and the number of instances are displayed. Finally, the fourth section identifies the underlying operating systems presently running for each SQL Server instance.

### 1.1.3. Generating a SQL Server Summary Report and Proposal

Additional information can be obtained by clicking on the Generate Report/Proposal link in the Actions pane of the Microsoft SQL Server Discovery node in the Microsoft Assessment and Planning Toolkit. Results are generated and presented in an Office Word document. In addition to the Word document, the toolkit prepares an Office Excel workbook containing more details about your SQL Server environment.

The Office Excel Workbook contains three tabs on the bottom left of the worksheet. Click on the Summary tab to find the assessment summary of your SQL Servers and the number of instances associated with each one. When you click on the Database Instances tab, the assessment results for your SQL Server database instances are unraveled in over 20 columns. Some of the extras provided under this tab include CPU type, the number of cores running, total amount of RAM and the SQL Server Service Pack version. The final tab, Components, uses more than ten columns to organize its results. The logical disk drive name, number of logical processors, and number of cores are just some of the additional information provided in this tab. In Table 1, you will find a comprehensive list of all the columns provided in the Office Excel workbook.

|  |  |
| --- | --- |
| * Computer Name of the Servers or workstations on which SQL Server is installed * SQL Server Instance Name * SQL Server Product Name * SQL Server Version Number * SQL Server Service Pack * SQL Server Edition * Clustered * Language * SQL Server Sub Directory * SQL Service State * SQL Server Start Mode * Current Operating System * Operating System Service Pack Level * Operating System Architecture Type * Number of Processors * SQL Server Instance Name * SQL Server Product Name * SQL Server Version Number * SQL Server Service Pack * SQL Server Edition | * Clustered * Language * SQL Server Sub Directory * SQL Service State * SQL Server Start Mode * Current Operating System * Operating System Service Pack Level * Operating System Architecture Type * Number of Processors * Number of Cores * Number of Logical Processors * CPU Type and Speed * System Memory (MB) * Logical Disk Drive Name * Logical Disk Size (GB) * Logical Disk Free Space (GB) * Machine Type * Number of Host Processors * SQL Server Component Name * SQL Server Component Version * SQL Server Component Service Pack |

Table1. Columns Presented in the SQL Server Discovery Excel Workbook.

**Note**: Values such as the System Memory and Logical Disk Free Space are derived instantaneously from the SQL Server computer when it is scanned. These values do not consist of maximums, minimums or averages.

In addition, the existing SQL Server assessment report also includes hardware readiness of existing SQL Servers for SQL Server 2008 and indicates whether they are capable of running SQL Server 2008. Having this information is useful if you plan on migrating to SQL Server 2008 during your SQL Server virtualization efforts.

## 1.2. Step 2: Understanding Hyper-V Maximum Configurations

When planning your SQL Server virtualization strategy it is beneficial to understand the maximum configurations with Windows Server 2008 R2 Hyper-V to reduce capacity planning and performance degradation issues during the implementation phase of the project. This section communicates the maximum configurations for virtual machines, a physical host running Hyper-V, and Hyper-V in a Failover Cluster environment.

### 1.2.1. Maximum Configurations for Virtual Machines

Each virtual machine, also known as a guest operating system, can support up to 4 virtual processors and 64 GB of memory. A maximum of 4 Virtual IDE disks are supported on each machine. The boot disk, commonly referred to as the startup disk must be attached to an IDE device and a virtual machine. Either a virtual hard disk or a physical hard disk can be used. A maximum of 256 SCSI disks are supported on each virtual machine. This is possible because each virtual machine can support a maximum of 4 virtual SCSI controllers and each controller can support a total of 64 disks. Each Virtual hard disk can support up to 2040 GB. Virtual SCSI disks are recommended over virtual IDE disks when implementing SQL Server virtual machines. However, for SQL Server workloads, pass-through disks, where a physical disk or LUN is mapped exclusively to a VM, provide the best performance from a disk perspective. For more information on SQL Server and pass-through disks review the following two whitepapers: [*High Performance SQL Server Workloads on Hyper-V*](http://sqlcat.com/whitepapers/archive/2010/05/27/high-performance-sql-server-workloads-on-hyper-v.aspx)(http://sqlcat.com/whitepapers/archive/2010/05/27/high-performance-sql-server-workloads-on-hyper-v.aspx) and [*Running SQL Server 2008 in a Hyper-V Environment - Best Practices and Performance Recommendations*](http://sqlcat.com/whitepapers/archive/2008/10/03/running-sql-server-2008-in-a-hyper-v-environment-best-practices-and-performance-recommendations.aspx) (http://sqlcat.com/whitepapers/archive/2008/10/03/running-sql-server-2008-in-a-hyper-v-environment-best-practices-and-performance-recommendations.aspx). Each SQL Server workload placed in a virtual machine should comply with the maximums.

### 1.2.2. Maximum Configurations for Servers running Hyper-V

The physical servers running Hyper-V in Windows Server 2008 R2 can have the following maximums: 64 logical processors, 8 virtual processors per logical processor, 512 virtual processors per server, 384 operating virtual machines on a server, and up to 1 TB of memory. Finally, there is no maximum imposed on physical network adapters for Hyper-V, virtual networks, and network switch ports for any server. Therefore, when placing multiple SQL Server virtual machines within a single Hyper-V host, you must ensure the total SQL Server workload fits within these boundaries.

### 1.2.3. Maximum Configurations for Hyper-V and Failover Clusters

When maintaining high availability with your SQL Server virtual machines it is imperative to understand the maximum configurations for running Hyper-V in a failover cluster configuration. Highly available servers running Hyper-V can have a maximum of 16 nodes for each cluster and 64 virtual machines running on each node. It is critical during the planning phase to determine the number of nodes in each cluster that will handle failover and maintenance tasks. One passive node is the minimum recommendation for failover; however, in a 16 node cluster more than one may be used. Be aware that the number of virtual machines running on each node can be compromised by the amount of memory being used by each machine, networking and storage bandwidth, and disk spindles which affects disk I/O performance.

With the inventory results obtained in Step 1 and your knowledge of maximum configurations for Windows Server 2008 R2 Hyper-V in Step 2, you are prepared to answer critical questions like:

* Will the Hyper-V maximum configurations impact my SQL Server virtualization strategy?
* Is it possible to virtualize all of my relational and data warehouse databases?
* Which SQL Server components should I virtualize?
* How many SQL Server computers, instances and databases am I currently running in my infrastructure?

## 1.3. Step 3: Determine SQL Server Workloads to Virtualize

This step will help you determine if the SQL Server computers, instances, and databases in your environment can be successfully virtualized without capacity and performance penalties. This is achieved by gaining intimate knowledge of the resource utilization required for all of your SQL Server workloads from a performance perspective. In addition, examine the specific technical requirements for SQL Server and the operating system and map them against any constraints in the virtualization technology. This will ensure the desired SQL Server workload fits within a virtual machine. Many architects skip this planning step and therefore, their opportunity to thoroughly understand their SQL Server resource utilization. As a result, they often end up placing too many SQL Server virtual machines on a single physical Hyper-V host causing performance degradation on all their SQL Server computers.

The use of performance tools to capture key performance metrics such as processor, memory, and disk utilization for the SQL Server candidates you plan on virtualizing is a good way to gain an understanding of your resource utilization and workloads. The metrics captured can also be valuable if you plan to consolidate SQL Server computers, instances and databases onto fewer virtual servers. For example, based on the results of the inventory and performance analysis, you may realize the CPU and Memory on many of your SQL Server computers are underutilized and decide to consolidate the computers during your virtualization project. Many tools are available to help you capture SQL Server performance metrics:

* **Utility Control Point** - The SQL Server Utility Control Point is a new feature included with SQL Server 2008 R2. The Utility Control Point centrally monitors instances of SQL Server as it collects configuration and performance information every 15 minutes from managed instances of SQL Server. Once data is collected from the managed instances, the SQL Server Utility dashboard and viewpoints in SQL Server Management Studio (SSMS) can be used to gain insight into the performance and resource utilization of SQL Server computers, instances and databases you plan on virtualizing, as illustrated in Figure 3. The dashboard yields information about: Instance CPU Utilization, Database Files Utilization, Storage Volumes Utilization and Computer CPU Utilization. With its ability to identify underutilized resources, the Utility Control Point also doubles as a tool for identifying consolidation opportunities. For more information on the Utility Control Point, view the following link: (<http://msdn.microsoft.com/en-us/library/ee210548(v=SQL.105).aspx>).

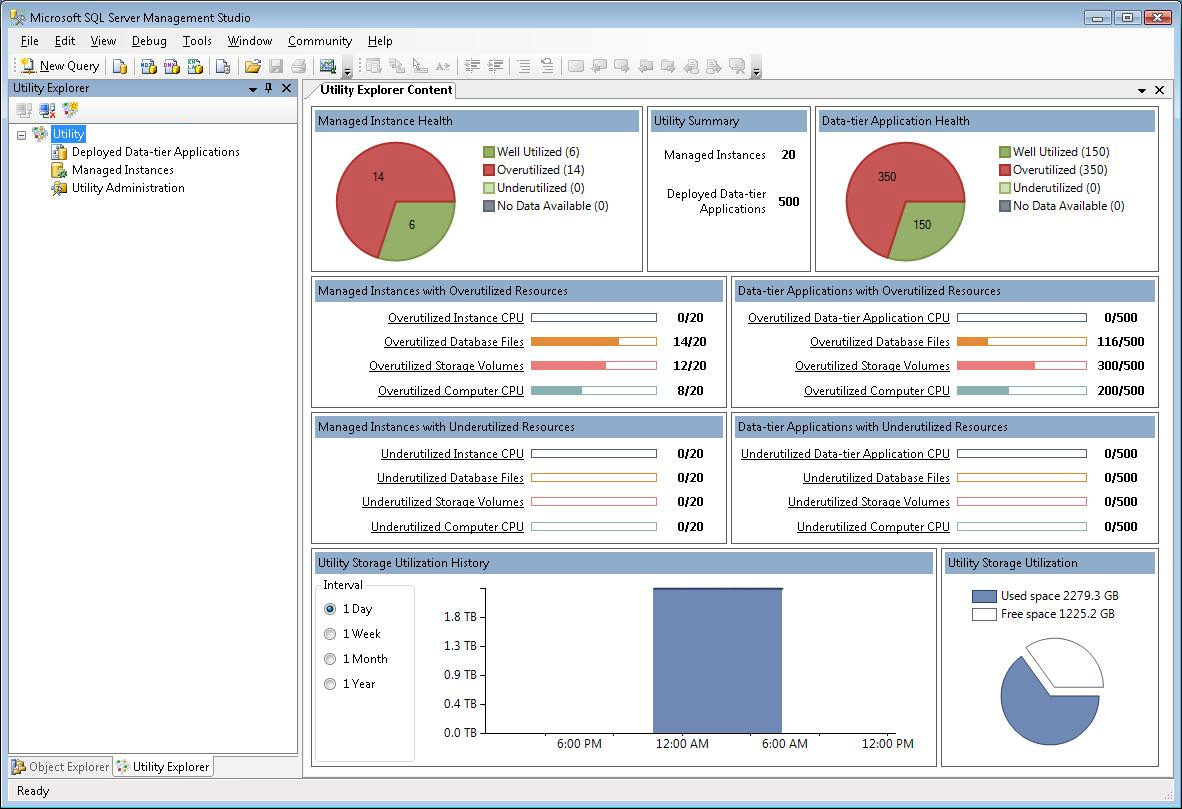


Figure 3. The SQL Server Utility Dashboard and Viewpoints

* **Data Collector and Management Data Warehouse:** The management data warehouse is a relational database containing performance data collected from a SQL Server instance that is a data collection target. Use of the built-in reports describing current and historical performance data will enable you to understand the workloads and resource utilization associated with your SQL Server computers, instances and databases planned for virtualization. These reports also call out the most poorly performing queries for each instance. The server activity, disk usage and query activity reports are just a few of the reports available to you for analysis. Review the following link for more information on the Management Data Warehouse: (<http://msdn.microsoft.com/en-us/library/bb677306.aspx>).
* **Performance Monitor** - Another way to obtain performance metrics is by using the Performance Monitor tool included in Windows Server 2008 R2 and Windows Server 2008. A tremendous amount of SQL Server and Windows performance counters are available to you with this tool. However, if you are seeking to capture information on the most pertinent performance objects and counters to assist with your virtualization efforts, then at the very least you will want to include the following:

|  |  |  |
| --- | --- | --- |
| **Performance object** | **Counter** | **Comments** |
| Memory | Pages/sec | Shows whether memory is paging to disk |
| Memory | Available Mbytes | Displays the amount of memory |
| Network Interface | Packets received/sec | Per each network card, especially ones being used for SQL Server |
| Network Interface | Packets sent/sec |
| Network Interface | Bytes received/sec |
| Network interface | Bytes sent/sec |
| Physical Disk | %Disk Read Time | Configure per disk as well as \_Total; will show how I/O looks at a physical level for the disk configuration |
| Physical Disk | %Disk Write Time |
| Physical Disk | %Idle time |
| Physical Disk | Avg. Disk Bytes/Read |
| Physical Disk | Avg. Disk Bytes/Write |
| Physical Disk | Avg. Disk Read Queue Length |
| Physical Disk | Avg. Disk Write Queue Length |
| Physical Disk | Avg. Disk sec/Read |
| Physical Disk | Avg. Disk sec/Write |
| Physical Disk | Split IO/sec |
| Process | % Processor Time | Per process (focus on the SQL Server processes, but you may want to gather all processes); gives information about the processes running on the server |
| Process | IO Read Bytes |
| Process | IO Read Operations |
| Process | IO Write Bytes |
| Process | IO Write Operations |
| Process | Page Faults/sec |
| Process | Thread count |
| Process | Working Set |
| Processor | % Processor Time | \_Total and per individual processor (especially with multiple cores); shows the performance of the processors |
| SQL Server:Databases (2000,2005,2008)\* | Transactions/sec | Select per database; tells how many transactions are happening in the database at that given time |
| SQL Server:General Statistics (2000) | User Connections | Gets the total amount of user connections in SQL Server |
| SQL Server:Memory Manager (2000, 2005, 2008) | Connection Memory(KB) | Displays the amount of memory consumed by connections |
| SQL Server:General Statistics (2008) | Active Temp Tables | Indication of tempdb usage (but not necessarily the only indicator) |

|  |  |  |
| --- | --- | --- |
| **Performance object** | **Counter** | **Comments** |
| SQL Server:Memory Manager (2000, 2005, 2008) | Target Server Memory (KB) | Displays the total amount of memory that can be used by the instance of SQL Server |
| SQL Server:Memory Manager (2000, 2005, 2008) | Total Server Memory (KB) | Displays the amount of dynamic memory being used by the instance of SQL Server |

* **Activity Monitor** - To gain quick insight into resource utilization for a SQL Server instance, use the Activity Monitor included in SQL Server 2008 R2 and SQL Server 2008. It includes a performance dashboard that provides graphs and information pertaining to Processor Time, Waiting Tasks, Database I/O and Batch Requests/sec, as illustrated at this link: (<http://msdn.microsoft.com/en-us/library/ms175518.aspx>).
* **Operations Manager** - Systems Center Operations Manager 2007 R2 includes a SQL Server and Windows Server Management Pack that will help gather performance data from SQL Server systems, instances, and databases you plan on virtualizing. Once the data is gathered, you can use Operations Manager to output intuitive reports to help you identify prime candidates for virtualization.

## 1.4. Step 4: Determine High Availability, Security and Isolation Strategies

Determining your high availability and security requirements are very important as they can influence the type of Hyper-V hardware you purchase and dictate where you place your SQL Server virtual machines.

### 1.4.1. High Availability Strategies

By choosing the appropriate high availability options, it is possible to meet your business continuity service level agreements, enable data protection and improve manageability of your SQL Server virtualized environment. Here are details of each high availability strategy:

#### 1.4.1.1. High Availability with Live Migration

A common SQL Server high availability strategy employed by many organizations is Live Migration. With Live Migration, running SQL Server virtual machines can be moved from one Hyper-V physical host to another Hyper-V physical host without service disruption or downtime. This is displayed in figure 4. As mentioned earlier, Live migration is integrated with Windows Server 2008 R2 Hyper-V and Microsoft Hyper-V Server 2008 R2 and requires the failover clustering feature to be added and configured on the servers running Hyper-V. By leveraging Live Migration, architects and database administrators can continue to ensure service continuity and maintain their SQL Server virtualization high availability objectives even during planned maintenance.



Figure 4. Achieving High Availability with Live Migration

#### 1.4.1.2. High Availability with Guest Clustering Using iSCSI

Another SQL Server high availability option involves guest clustering using iSCSI. Guest failover clustering, as illustrated in Figure 5 makes the provisioning of clusters possible in a virtual environment. A guest failover cluster is simply SQL Server failover clusters inside virtual machines with nodes operating as virtual machines. SQL Server 2005, SQL Server 2008 and SQL Server 2008 R2 support guest failover clustering. The shared storage required to build a cluster is connected to each virtual machine by means of iSCSI. SQL Server guest cluster nodes can all run on the same physical host machine or on different physical host machines. If SQL Server guest clustering is running on the same host, business operations and high-availability will be compromised if the host becomes unavailable. Therefore, consider running the active node and standby node of a SQL Server guest cluster on different physical host machines to maintain high availability and reduce interruptions in business operations with your SQL Server virtual machines.



Figure 5. Achieving High Availability with Guest Clustering

#### 1.4.1.3. Combining High Availability Alternatives

Database Mirroring, Log Shipping or Peer-to-Peer replication can be leveraged in conjunction with Live Migration or guest clustering using iSCSI. By combining the high availability alternatives you can maximize availability, business continuity, disaster recovery, and manageability of your SQL Server virtualized environment. For example, if you implemented a SQL Server virtual machine on a guest cluster in San Francisco, California and another SQL Server virtual machine on a guest cluster in Toronto, Ontario, you can implement Database Mirroring or Log Shipping as a disaster recovery mechanism to ensure business continuity between the SQL Server virtual machines in the two different locations. Moreover, you may combine high availability with virtualization by implementing a worldwide peer-to-peer replication topology where every single SQL Server instance partaking in the replication topology is virtualized.

### 1.4.2. Security and Isolation Strategies

Understanding the security and isolation requirements can not only influence where you store your virtual machines, but can influence whether or not SQL Server computers, instances, and databases can be consolidated.

For example, strict security requirements may be in place for the Human Resources department to ensure their databases are isolated from other production databases. These types of databases may require isolation as they are governed by a specific regulatory compliance. Finally, the TempDB database may be a performance bottleneck if you consolidate too many databases onto a single virtual instance of SQL Server.

To achieve complete application and operating system isolation when virtualizing SQL Server, you can maintain different versions and editions of SQL Server within the same physical Hyper-V host. For example, you can virtualize SQL Server 2008 R2, SQL Server 2008, SQL Server 2005 and SQL Server 2000 on separate virtual machines running on the same Hyper-V host. If needed, you can also maintain different editions such as 32-bit editions and 64-bit editions of the application and operating system.

## 1.5. Step 5: Determine the Version and Edition of SQL Server and the Operating System

The SQL Server version, edition, and operating system you choose for your SQL Server virtual machines is based on numerous conditions including whether or not you plan to use the Physical-to-Virtual (P2V) tool, guest clustering or consolidate. For example, if your plans include the use of the P2V conversion tool then you will be migrating your physical servers to virtual servers "as is". As such, the version and edition of your SQL Server after conversion remains the same. On-the-other-hand, if you plan on deploying a brand new SQL Server virtual machine from scratch, you will have to determine the SQL Server version and Operating System you will use.

Your decision can also be impacted by the SQL Server features or components you plan on deploying. If your plan involves Guest Clustering then the Enterprise edition of Windows Server 2008 R2 is required. Likewise, if you plan on managing more than 25 instances of SQL Server with a SQL Server Utility Control Point then the Datacenter edition of SQL Server 2008 R2 is required.

It is beneficial to review the following links to fully understand which version and edition of SQL Server and Windows Server you require based on the features you plan on implementing:

### 1.5.1. SQL Server Editions, Components and Features

* SQL Server 2008 R2 Editions and Components (<http://www.microsoft.com/windowsserver2008/en/us/r2-editions-overview.aspx>)
* Features Supported by the Editions of SQL Server 2008 R2

(<http://msdn.microsoft.com/en-us/library/cc645993(SQL.105).aspx>)

* SQL Server 2008 Editions and Components

(<http://msdn.microsoft.com/en-us/library/ms144275.aspx>)

* Features Supported by the Editions of SQL Server 2008

(<http://msdn.microsoft.com/en-us/library/cc645993.aspx>)

### 1.5.2. Windows Server Editions and Features

* Windows Server 2008 R2: Overview of Features and Editions

(<http://www.microsoft.com/windowsserver2008/en/us/r2-editions-overview.aspx>)

## 1.6. Step 6: Determine Virtualization Strategy

Three strategies are available when migrating your SQL Server environment from physical servers to virtual servers. The first option is using a conversion tool to execute operations *"as is.”* The second option is deploying a new virtualized SQL Server instance and conducting a database migration from the legacy physical server. During a SQL Server virtualization strategy, many organizations take the opportunity to upgrade their databases to the latest database platform or consolidate their SQL Server instances and databases onto fewer virtualized systems reducing costs and simplifying management. As a result, consolidation through virtualization is the third option.

### 1.6.1. Virtualization Strategy: Physical-to-Virtual Conversion

Microsoft System Center Virtual Machine Manager (VMM) 2008 R2 can be used to convert existing physical SQL Server computers into virtual machines through a process commonly referred to as physical-to-virtual (P2V) conversion. VMM simplifies P2V conversions with the use of a task-based wizard that automates much of the conversion process. Since the P2V process is completely scriptable, you have the ability to initiate large-scale P2V conversions of SQL Server through the Windows PowerShell command line. It is worth mentioning that the conversion from SQL Server physical servers to virtual servers in this strategy is accomplished through a one-to-one mapping process. As a result, while you will see a decrease in the number of physical servers in your infrastructure, you will still have the same number of SQL Server instances to manage.

Let's say your environment includes one physical SQL Server instance running SQL Server 2000 (named SQLServer1) that hosts two databases and another physical SQL Server instance running SQL Server 2005 (named SQLServer2) with two databases. In this example, using the P2V tool may not be your best option as you will be reducing the physical footprint by virtualizing the two physical servers; however, SQL Server management is not simplified as two servers still remain after the P2V conversation. If SQLServer1 and SQLServer2 are underutilized, virtualization through SQL Server instance and database consolidation may be a better choice. For more information on SQL Server consolidation strategies and best practices, review the following white paper: *Consolidation Using SQL Server 2008*, at (http://msdn.microsoft.com/en-us/library/ee819082.aspx).

### 1.6.2. Virtualization Strategy: Server Consolidation

The other Virtualization Migration strategy consists of implementing a new virtualized SQL Server instance and then migrating the data from the source physical server. Let's consider the previous example. Instead of conducting a P2V conversion from the source instance running SQL Server 2000 (SQLServer1), you can choose to implement a new virtualized SQL Server instance (SQLServer3) running SQL Server 2008 R2 and then conduct a migration of the data from the source server (SQLServer1), to the new target virtualized SQL Server instance (SQLServer3). By implementing a new virtual machine with SQL Server 2008 R2 and then conducting a migration, you have the capability to upgrade the legacy database platform to the latest version of SQL Server.

Alternatively, you can conduct a P2V migration of an existing legacy SQL Server instance and then conduct an in-place upgrade after it has become virtualized. As you can see the possibilities are endless.

## 1.7. Step 7: Determine Hyper-V Hardware

The planning phase concludes when the hardware requirements to support your SQL Server virtualization environment are determined. The hardware required will likely be dictated by several predetermined factors in the planning phase such as the total amount of processor, memory, storage and network bandwidth required. Factors such as high availability, security, management, isolation requirements and the underlying operating system determined during the planning phases will also dictate the type of hardware purchased. For example, if you have already decided on implementing the Live Migration feature for high availability then more than one Hyper-V host will be needed and you will have to configure a failover cluster. Furthermore, based on the total amount of storage required, you may use local storage, direct attach storage, or even a SAN. As you may already realize, there isn't one ‘silver bullet’ hardware reference architecture that will meet every solution. The Hyper-V hardware will be determined based upon your SQL Server virtualization workloads you plan on supporting. Don't forget to take future growth into consideration when planning capacity. Let's evaluate some hardware strategies to assist in the decision making process:

### 1.7.1. Hyper-V R2 Enhancements

Windows Server 2008 R2 comes equipped with many new capabilities and enhancements to Hyper-V virtualization that provide improved flexibility, increased performance, simplified management, and greater scalability for SQL Server virtual machines, including the following:

* **Live migration** - Live migration is an advocate of high availability. It makes it possible to move running virtual machines from one node of the Hyper-V failover cluster to another node in the same cluster without any interruption in service.
* **Cluster Shared Volumes (CSV)** - CSV provides increased reliability when used with Live Migration and virtual machines, and also provides a single, consistent file namespace that allows all servers running Windows Server 2008 R2 to see the same storage.
* **Dynamic virtual storage support** - The support for hot plug-in and hot removal of the storage is a major enhancement. Virtual machines can be configured and reconfigured numerous times to fulfill changes in requirements by either adding or removing virtual hard disks and physical disks on a running virtual machine.
* **Enhanced processor support** - You can now have up to 32 physical processor cores, increasing your ability to take on more workloads that are more challenging on a single host. In addition, Second-Level Address Translation (SLAT) and CPU Core Parking are supported and should be utilized when procuring hardware. With CPU Core Parking, Windows Server 2008 R2 and Hyper-V can consolidate processing resulting in a smaller number of processor cores and a reduction in power consumption. Inactive processor cores are put to "sleep" until needed. SLAT adds a second level of paging below the architectural x86/x64 paging tables in x86/x64 processors. It provides an indirection layer from virtual machine memory access to the physical memory access. In virtualization scenarios, hardware-based SLAT support improves performance.
* **Enhanced networking support** - Support for jumbo frames has been extended to virtual machines. If the underlying physical network will support it, virtual machines can use jumbo frames up to 9,014 bytes in size.

For more details on the Windows Server 2008 R2 Hyper-V enhancements and changes, see: (<http://technet.microsoft.com/en-us/library/dd391932(WS.10).aspx>).

### 1.7.2. Storage - Pass-through vs. Fixed Size VHD

The best practice for SQL Server workloads that are I/O intensive is to choose pass-through disks instead of fixed disks. A fixed disk is a virtual hard drive (VHD), whereas a pass-through disk is exclusively assigned to the VM. Through repeated tests of various random and sequential I/O, the SQL Server Customer Advisory Team (CAT) found the I/O overhead of Hyper-V using pass-through disks is from none to minimal. In addition, do not use Dynamic disks with VMs as you will suffer performance degradation when the disk grows due to increased I/O loads.

### 1.7.3. Hyper-V R2 System Requirements

Independent of the SQL Server workloads you plan on virtualizing you have to ensure that each Hyper-V physical host also meets the minimum system requirements. The following Table illustrates the system requirements:

|  |  |
| --- | --- |
| **Component** | **Specifications** |
| Processor | * Requires 64-bit machines with AMD-V or Intel Virtualization Technology enabled processors * Supports single and multi-core x64 processors * 1.4 GHz; Recommended: 2 GHz or faster * Hyper-V server does not support Itanium systems |
| Memory (RAM) | * Minimum: 1 GB RAM * Recommended: 2 GB RAM or greater * Additional RAM is required for each SQL Server guest operating system * Maximum 1 TB |
| Disk Space | * Minimum: 8 GB * Recommended: 20 GB or greater * Additional disk space needed for each SQL Server guest operating system |
| SQL Server Related Operating Systems Supported | * Windows Server 2008 R2 - VM configured with 1, 2, or 4 virtual processors (Standard, Enterprise, Datacenter and Web) * Windows Server 2008 x64 Edition - VM configured with 1, 2, or 4 virtual processors (Standard, Enterprise, Datacenter, Web) * Windows Server 2008 x86 - VM configured with 1, 2, or 4 virtual processors (Standard, Enterprise, Datacenter, Web) * Windows Server 2003 x86 w/ Service Pack 2 - VMs configured with 1 or 2 virtual processors (Standard, Enterprise, Datacenter, Web) * Windows Server 2003 R2 x86 w/ Service Pack 2 - VMs configured with 1 or 2 virtual processors (Standard, Enterprise, Datacenter, Web) * Windows Server 2003 R2 x64 w/ Service Pack 2 - VMs configured with 1 or 2 virtual processors (Standard, Enterprise, Datacenter) * Windows Server 2003 x64 Edition w/ Service Pack 2 - VMs configured with 1 or 2 virtual processors (Standard, Enterprise, Datacenter) * Windows Server 2000 w/ Service Pack 4 - VMs configured with 1 virtual processor (Server and Advanced Server) |

# 2. Implementing SQL Server Virtualization

When all of the steps in the planning phase are complete and the new Hyper-V hardware is procured you are ready to implement your SQL Server virtualization strategy. This section opens with a discussion on the Physical to Virtual (P2V) migration tool located in System Center Virtual Machine Manager 2008 R2 and its availability. Before being given step by step instructions for carrying out a P2V conversion, you will have insight of the P2V conversion process.

## 2.1.1. Operating Systems Supported

P2V migrations can be carried out using an online or offline conversion method. The method you use is largely determined by the operating system currently running on the physical SQL Server computer you plan on converting and its ability to support either an online or offline conversion. Operating systems currently supporting online or offline conversions are listed below:

* Windows Server 2003 operating systems with Service Pack 1 (SP1) or above
* Windows Server 2003 R2 Standard Edition operating system or above
* Windows Server 2008
* Windows Server 2008 R2
* Windows XP operating systems with SP1 or above
* Windows Vista
* Windows 7

Furthermore, be aware of the following operating systems if you are considering a P2V conversion, as they do not support P2V conversions:

* Windows Server 2003 R2 Standard x64 Edition operating system
* Windows XP Professional x64 Edition operating system
* Microsoft Windows NT Server 4.0 operating system

Also, note that an online conversion of Windows Server 2000 Server Service Pack 4 is not supported; however, an offline conversion can be conducted.

## 2.1.2. P2V Conversion Process

The online P2V conversion process is straightforward. Below are 5 steps to help you discern the process:

1. The process begins with installing a Virtual Machine Manager agent on the physical SQL Server computer you plan on virtualizing.
2. Once the conversion is completed in Step 6, the agent is removed.
3. The agent collects hardware and software data, services, hotfixes, and volume configuration on the physical SQL Server computer being converted. The information collected is exported and stored on the VMM database until it can be analyzed to determine if virtualization of the physical SQL Server is achievable.
4. If the physical SQL Server can be virtualized then imaging is completed. The Volume Shadow Copy Service (VSS) is used to capture an image of each NTFS volume. By leveraging Background Intelligent Transfer Service (BITS), a separate virtual disk with the extension .vhd is created for each physical volume.
5. Final preparation of the virtual hard disks and virtual machine takes place.
6. The conversion process draws to a close with the creation of a new virtual machine based on the physical SQL Server computer and attachment of the virtual hard disks, network adapters, processors, memory and input devices.

**Note**: The P2V tool has built-in intelligence to make the VM highly available after conversion if the P2V physical host target is already clustered. This provides low-cost high availability for SQL Server. The SQL Server instance itself will not be clustered so failover only occurs at the host for the guest VM process.

## 2.1.3. P2V Conversation Step-by-Step

Once you are ready to carry out the SQL Server physical to virtual conversion with Virtual Machine Manager (VMM) 2008 R2, proceed with the steps below:

1. On the computer running Virtual Machine Manager 2008 R2 click Start, All Programs, Microsoft System Center, Virtual Machine Manager 2008 R2 and then select Virtual Machine Manager Administrator Console.
2. In the Virtual Machine Manager Administrator Console select Actions, Virtual Machine Manager and then Convert Physical Server to launch the wizard (see Figure 6).
3. On the Select Source page, configure the parameters below:

* Computer name or IP address: Identify the SQL Server physical computer that will eventually be used as the new virtual machine by either typing the computer name or IP address, or clicking Browse to locate the server.
* User name: Provide a user account name with local administrator permissions on the physical server.
* Password: Provide the password for the local administrator account selected.
* Domain or Computer Name: Enter the domain name for the user account you plan on using. If the physical server is not associated with a domain then use a local user account.

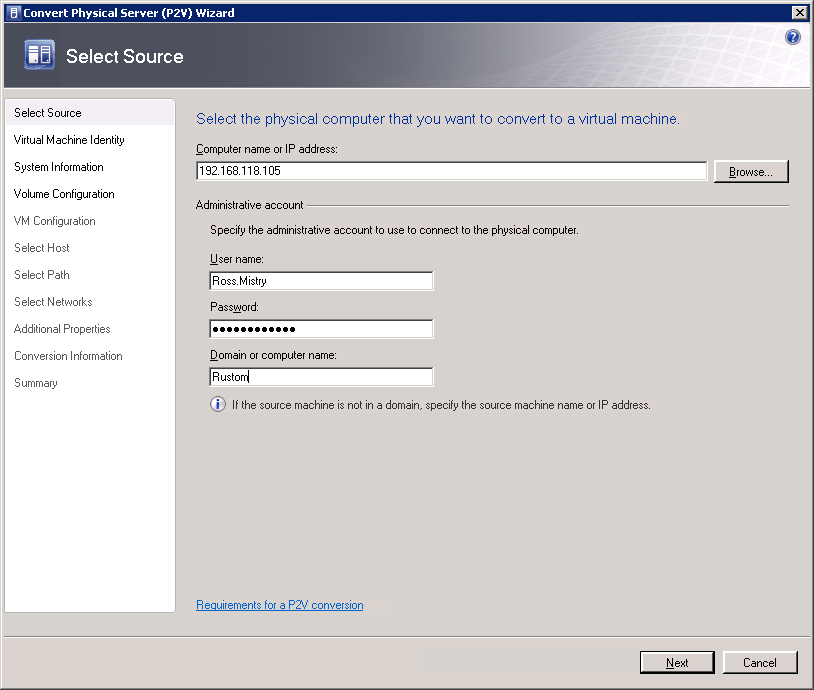


Figure 6. Select the SQL Server physical computer you want to convert to a virtual machine

1. On the Virtual Machine Identify page, provide the Virtual Machine Name, Owner and Description for the Virtual Machine which is optional and then click Next to continue. Note, the virtual machine name selected identifies the virtual machine for systems administration on System Center Virtual Machine Manager 2008 R2. As such, the name does not have to match the computer name of the physical computer; however, it is beneficial to use the same name for simplicity. For the owner value, you can either accept the pre-populated value, which is DomainName\Username, or you can select an Active Directory account of your choice as the owner by clicking on the Browse button.
2. Click Scan System on the System Information page (see Figure 7). This action allows an agent to be temporarily installed on your SQL Server physical computer to collect system information. In addition, the wizard ensures the WMI service is installed and running for the scan to successfully work. Review the results to ensure your system is ready for a P2V conversion. Click Next to continue.

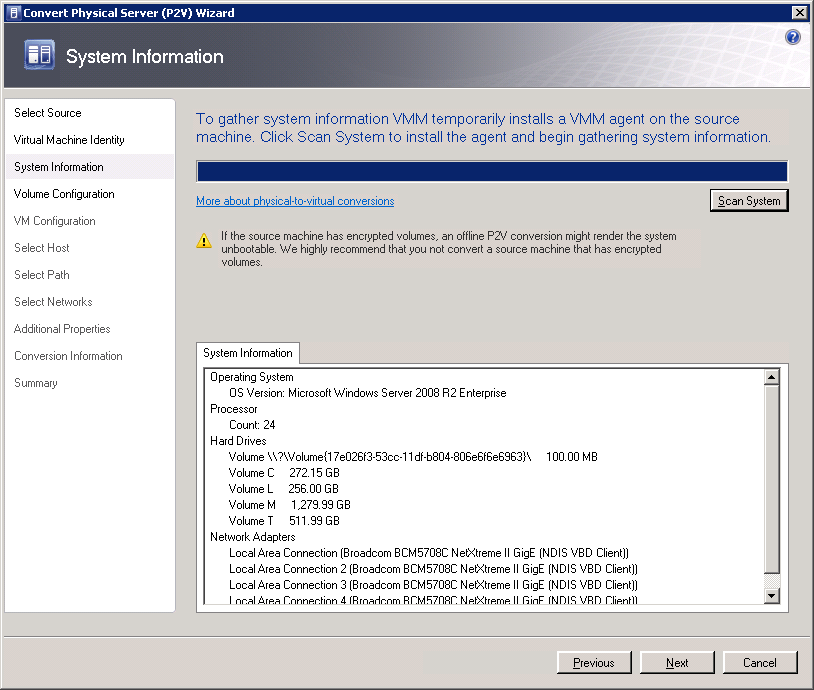


Figure 7. System Information Results

1. On the Volume Configuration page (see Figure 8), review the lists of all volumes displayed by the wizard and make appropriate changes. First, select only the system volumes and boot volumes on the source SQL Server machine that you want duplicated on the virtual machine. Second, for each volume selected adjust the settings to reflect your needs. For instance, specify VHD size, VHD type as dynamic or fixed, and the channel for any volume to which you will attach the .vhd. The third area on the Volume Configuration page is Conversion Options. If your SQL Server physical computer cannot support an online conversion, the offline conversion option is selected automatically. When the P2V conversion is successfully completed, click in the checkbox next to Turn off Source Computer after Conversion. Click Next to continue to the next page.

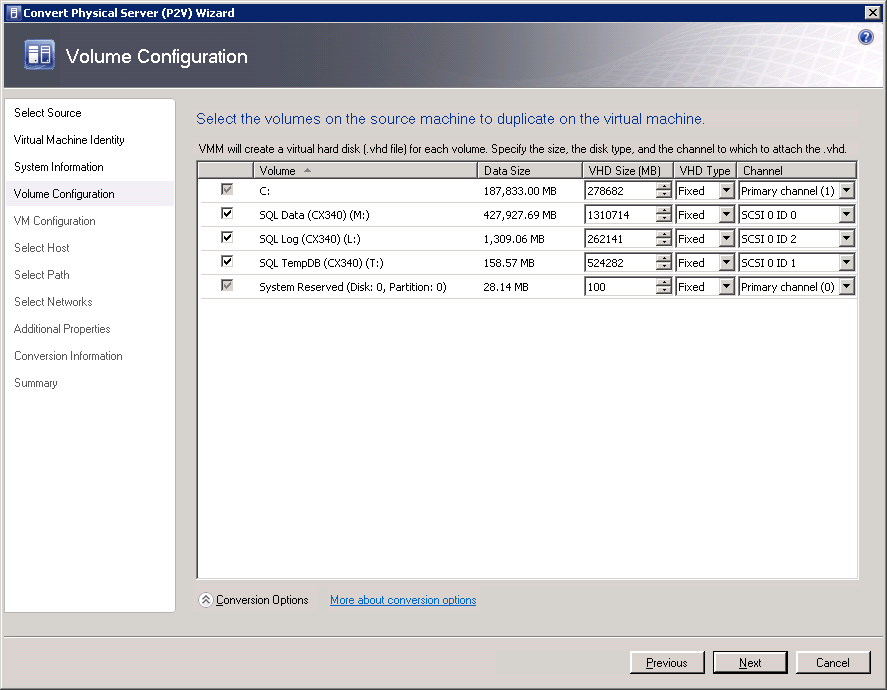


Figure 8. Selecting the Volumes on the Source Machine to Duplicate on the Virtual Machine

To maximize disk performance from an I/O perspective, Fixed disks are recommended for production SQL Server instances. The following bullets are an example of a disk configuration layout for a virtual SQL Server instance:

* Controller 1 Channel 0: Fixed VHD D:\ Binaries (SQL Install)
* Controller 1 Channel 1: Fixed VHD H:\ DATA
* Controller 1 Channel 2: Fixed VHD I:\ DATA (optional)
* Controller 1 Channel 3: Fixed VHD J:\ DATA (optional)
* Controller 1 Channel 4: Fixed VHD K:\ DATA (optional)
* Controller 2 Channel 0: Fixed VHD O:\ Log Drive
* Controller 3 Channel 0: Fixed VHD T:\ Tempdb Drive
* Controller 4 Channel 0: Fixed VHD E:\ Backup Drive

1. The Virtual Machine Configuration page turns its focus to the new virtual server. Specify the number of processors needed for the new virtual server along with the amount of memory necessary for successful operation. For example, for a large SQL Server workload running Windows Server 2008 R2, you may select 4 processors and 64 GB of RAM.
2. On the Select Host page (see Figure 9), specify the Hyper-V host for your new SQL Server virtual machine and then click Next. To help you determine the best Hyper-V host, look under Host Name and Rating to gain insight into how each host rated against the forecasted resource usage needs, and software and hardware needs of your virtual machine. Hosts receive a rating of zero to five stars, with 5 being the best. Additional information about each host can be found by clicking on the following tabs: Details, Rating Explanation, and SAN Explanation.

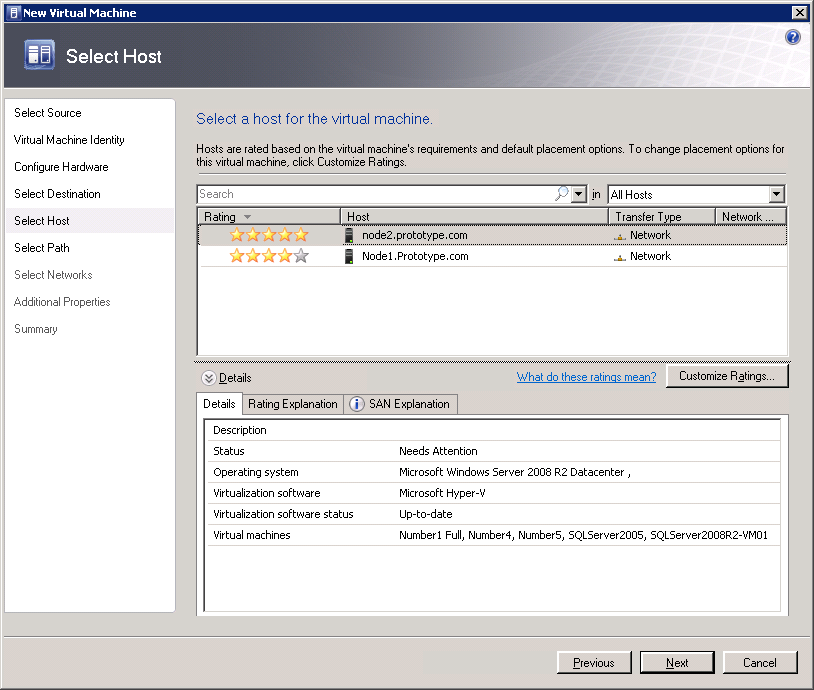


Figure 9. Select a Host for the SQL Server Virtual Machine.

1. On the Select Path page, save files to the host you selected by accepting the default path. Otherwise, click Browse to select a different path. Click Next to continue.
2. On the Select Networks page, assign a virtual network to each Physical Network Adapter found on the physical SQL Server computer you are converting. Click Next to continue.
3. On the Additional Properties page, specify the start and stop actions for your virtual machine and then click Next. The actions include:

* Action When Physical Server Starts
  + Never Automatically Turn on the Virtual Machine
  + Always Automatically Turn on the Virtual Machine
  + Automatically turn on the Virtual Machine if it was Running When Physical Server Stopped
* Action when Physical Server Stops
  + Save State
  + Turn off Virtual Machine
  + Shut down Guest OS

1. Review the details on the Conversion Information page and confirm there are no issues to prevent the physical to virtual conversion. If issues are detected, they must be resolved before proceeding with the P2V conversion. Click Check Again to rescan the computer when you have resolved each issue. If no issues are detected by the scan then click Next to continue.
2. Review the settings on the Summary screen and then click Create to begin the conversion. Alternatively, you can select the option to Start the Virtual Machine after deploying it on the host or click View Script which will enable you to view and copy the Windows PowerShell script responsible for operating the wizard and executing the P2V conversion.
3. Once in the Jobs window, you can monitor how the conversion is proceeding and confirm whether or not the conversion process is successful. Any jobs that fail will be identified in this window along with an error message and a recommended fix.

## 2.1.4. Starting and Stopping a SQL Virtual Machine

You can start or stop a SQL Server virtual machine by using either the Virtual Machine Manager Console included in System Center Virtual Machine Manager R2 or by using the Hyper-V management console included in Windows Server 2008 R2. The start and stop action are executed by performing the same operation regardless of the console you choose. Right-click on the SQL Server virtual machine of your choice and select either Start or Stop.

# 3. Implementing Live Migration for SQL Server 2008 R2 Virtual Machine

Before implementing Live Migration for one or more SQL Server virtual machines, several prerequisites must be satisfied. This section provides a quick walk through of the implementation process for Live Migration that includes the prerequisite steps and additional resources to assist you with your implementation.

Here are the high level steps to implement Live Migration:

1. A Hyper-V failover cluster must be implemented with Windows Server 2008 R2 before leveraging Live Migration. The hardware features selected for each node in the Hyper-V failover cluster, including the shared storage, must be supported by Windows Server 2008 R2. Your failover cluster is supported when all of the hardware features are marked, "Certified for Windows Server 2008 R2" and the configuration passes each test in the Validate a Configuration wizard. Consult the following link: (<http://go.microsoft.com/fwlink/?LinkId=139145>) for detailed information on hardware compatibility for Windows Server 2008 R2.
2. Install the Hyper-V role and Failover Clustering feature on each node within the failover cluster. Once this is completed, create a virtual network to support the SQL Server virtual machines, validate the cluster, and create the failover cluster. For step-by-step instructions on configuration, review the *Hyper-V: Using Hyper-V and Failover Clustering Guide* at: <http://technet.microsoft.com/en-us/library/cc732181(WS.10).aspx#BKMK_Install>).
3. Configure Cluster Shared Volumes by performing the steps in section 3.1 of this guide.
4. Setup a virtual machine for Live Migration by performing the steps in section 3.2 of this guide.
5. Commence a Live Migration of a virtual machine. See section 3.3 of this guide for instructions.

Note that for VMs created through P2V migration, if the physical host that the VM is migrated to is already clustered, then the VM will be automatically configured to be highly available and ready for Live Migration so the steps above are not needed.

## 3.1. Enabling Cluster Shared Volumes (CSV)

Although it is not required in a Windows failover cluster, it is beneficial to take advantage of Cluster Shared Volumes for use with Hyper-V when deploying a Live Migration. The Cluster Shared Volumes feature is available with some versions of failover clustering in Windows Server 2008 R2. Its goal is to simplify management by allowing multiple VMs to be stored on a single Logical Unit Number (LUN) reducing storage challenges as the number of LUNs required to support VMs drastically reduces. With CSV, clustered virtual machines are able to failover independently of one another eliminating one of the major issues with the last version of failover clustering. Lastly, CSV also decreases the chance of experiencing service disruption when your migration is completed as the disk resource does not have to be moved, unlike a standard cluster disk.

You can enable Cluster Shared Volumes with the Failover Cluster Manager when running a Hyper-V cluster on Windows Server 2008 R2. Here are the steps you need to perform:

1. Launch Failover Cluster Manager by clicking Start, Administrative Tools and then Failover Cluster Manager.
2. In the Failover Cluster Manager snap-in, click on the name of the cluster you wish to enable in the console tree. If the cluster name is not displayed, click on Failover Cluster Manager and then click on Manage a Cluster. Either select the cluster or enter the name of a cluster.
3. Right-click on the failover cluster and then click Enable Cluster Shared Volumes.
4. The Enable Cluster Shared Volumes dialog box is invoked (see Figure 10). Read and accept the terms and restrictions associated with using Cluster Shared Volumes and then click OK.

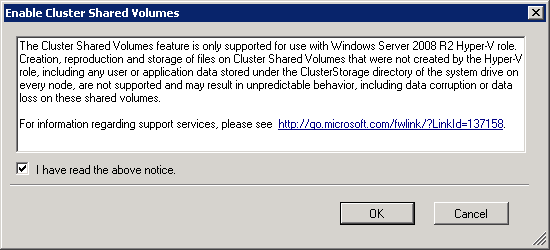


Figure 10. Enable Cluster Shared Volumes Dialog Box.

1. Add storage to the Cluster Shared Volumes by clicking on Cluster Shared Volumes and selecting Add Storage or Add Storage under Actions.
2. In the Add Storage dialog box (see Figure 11) a list of available disks is displayed. Select a disk or disks from the list and click OK.

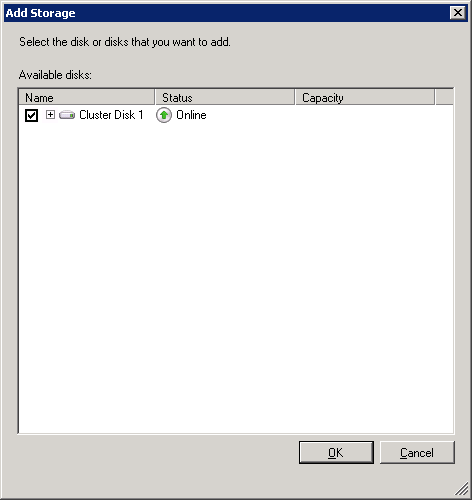


Figure 11. Adding Storage to a Cluster Shared Volume.

1. Once the disk or disks selected are added, they show up in the Results pane for Cluster Shared Volumes (see Figure 12).

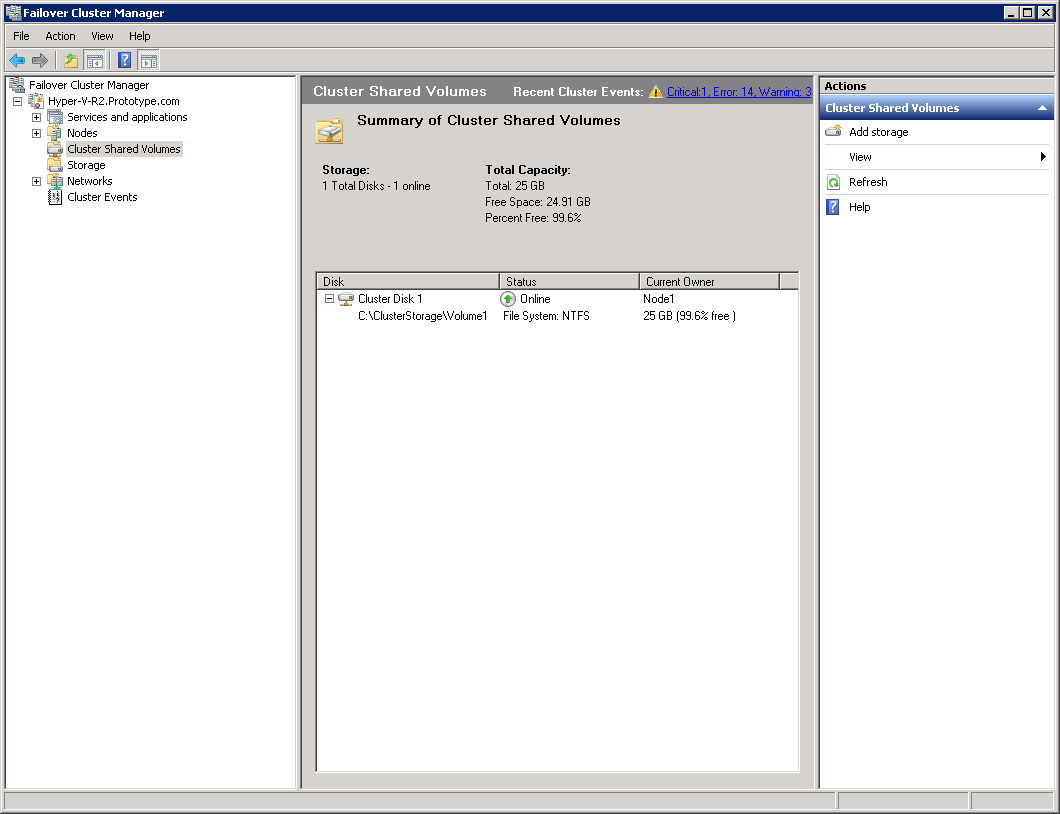


Figure 12. Viewing the Cluster Shared Volumes in Failover Cluster Manager.

**Note:** The storage location for Cluster Shared volumes is found under systemRoot\ClusterStorage\NameofVolume. For example, a CSV called Volume 1 will be presented to all nodes within the Hyper-V failover cluster as C:\ClusterStorage\Volume1. Folders created for each volume on the disk(s) are added to the Cluster Shared Volumes and stored in this location.

## 3.2. Creating a SQL Server Virtual Machine with Hyper-V

Another critical step is creating high availability for the new SQL Server virtual machine you setup. You can accomplish this task with either the Hyper-V Manager Console or the Virtual Machine Manager (VMM) 2008 R2 Administrator Console. Those using the Hyper-V Manager Console are taken through a three phase process that involves creating the VM, reconfiguring the automatic start action for the VM, and finally achieving a highly available virtual machine. When using Virtual Machine Manager 2008 R2, the installation process is streamlined as the three phase process initiated by the Hyper-V Manager Console is rolled up together with the New Virtual Machine wizard. Moreover, the Virtual Machine Manager 2008 R2 Administrator Console offers additional benefits including the ability to use an existing virtual machine template, import settings from a hardware profile, and intelligent virtual machine placement. The next two subsections walk you through the steps for creating a Virtual Machine with the Hyper-V Manager Console and also the Virtual Machine Manager 2008 R2 Administrator Console.

### 3.2.1. Creating a SQL Server Virtual Machine with Hyper-V Manager

Carry out the instructions below to create a virtual machine with the Hyper-V Manager Console. The steps are separated into three phases.

**Phase 1: Create a Virtual Machine**

1. Launch Hyper-V Manager by clicking Start, Administrative Tools, and then Hyper-V Manager.
2. Launch the New Virtual Machine wizard (see Figure 13) from the Actions pane by selecting New, and then clicking Virtual Machine.
3. Read the information on the Before You Begin page and then click Next. You can select the option to hide this page when using the wizard in the future.
4. On the Specify Name and Location page, enter the name of the SQL Server virtual machine. Next, select the option Store the virtual machine in a different location. Either type the location where the virtual machine will be stored or use the browse button to find the location. Click Next. To take advantage of Live Migration, store the SQL Server Virtual Machine on a Cluster Shared Volume as displayed in Figure 13.

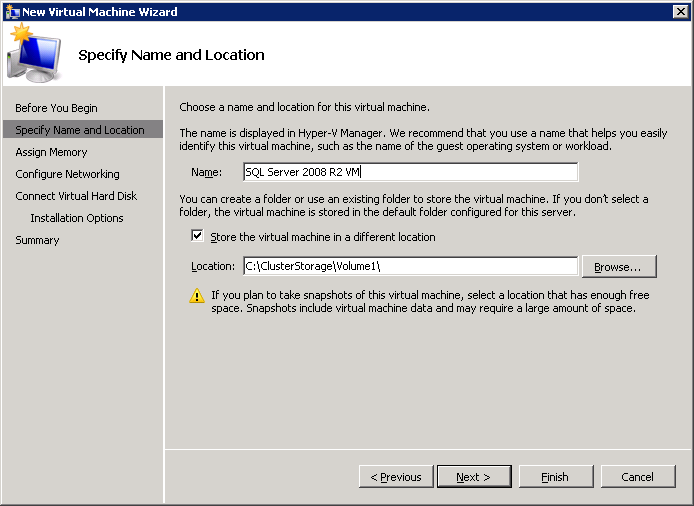


Figure 13. Creating a new SQL Server Virtual Machine.

1. On the Memory page, allocate the amount of memory in megabytes (MB) for the SQL Server virtual machine's guest operating system. This number will be based on your findings in the planning phase which is covered in Part 1 of this whitepaper. More RAM can always be added in the future. Click Next to continue to the next page.
2. On the Networking page, connect the network adapter to an existing virtual network by selecting the appropriate network adapter from the drop down menu. Click Next.
3. On the Connect Virtual Hard Disk page you can create a virtual hard disk by specifying its name, location, and size. Alternatively, you can choose an existing virtual hard disk or attach a virtual hard disk at a later time. For the purpose of the following steps, create a new virtual hard disk and ensure the virtual hard disk resides on a Cluster Shared Volume. Click Next to continue.
4. On the Installation Options page (see Figure 14), select Installing an operating system from a boot CD/DVD-ROM and then specify the Media location under Media. Click Next. You can select other options for installing an operating system that include:

* Installing an operating system at a later time
* Installing an operating system from a boot floppy disk
* Installing an operating system from a network-based installation server

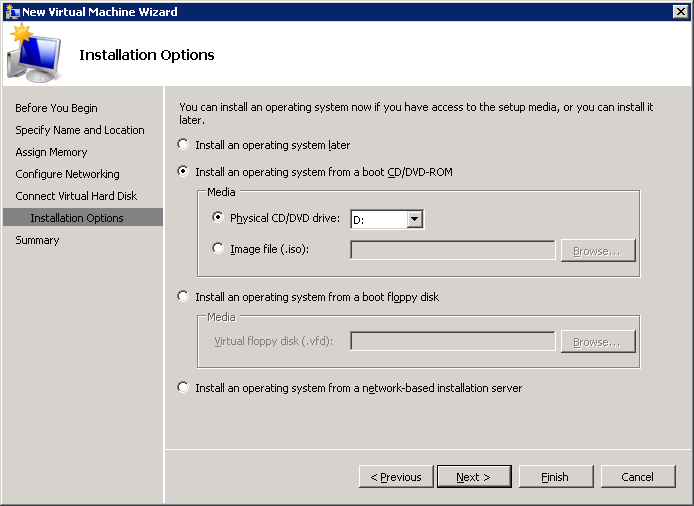


Figure 14. Installing the Operating System for a new SQL Server Virtual Machine.

1. On the Completing the New Virtual Machine Wizard page, review your selections and then click Finish.
2. The new Virtual Machine folder is created along with the virtual machine configuration file and virtual disks; however, it will be in an offline state. To make your virtual machine highly available it must remain in this state.
3. In Hyper-V Manager, click on the name of the SQL Server virtual machine you created in the Virtual Machines section of the results pane. Click Connect. The Virtual Machine Connection tool opens.
4. From the In the Action menu in the Virtual Machine Connection window, click Start.
5. Follow the prompts to select and install a Windows operating system.
6. As part of your setup, install Hyper-V Integration Services tools. Connect to the virtual machine and then select Insert The Integration Services Setup Disk from the Action Menu of the Virtual Machine Connection window. Next, click Install on in the AutoPlay dialog box to install the tools. It is a best practice to install the Hyper-V Integration Services tools for every VM created. This software package provides Virtual Server Client (VSC) code, which is responsible for increasing the Hyper-V performance of the virtual machine (VM) from an I/O, memory management, and network performance perspective.

**Note:** Disregard step 14 if you are running the Windows Server 2008 R2 Operating System as the Hyper-V Integration Services tool is already included with the Operating System.

1. If you plan on using Pass through disks for SQL Server, now is the time to mount and configure the drives with the ISCSI initiator and Disk Management console included with Windows Server 2008 R2.
2. Once the Operating System, Hyper-V Integration Services and Pass through disks are successfully installed and configured, you are ready to install SQL Server. Review the link for planning and installing SQL Server 2008 R2, found here: (<http://technet.microsoft.com/en-us/library/bb500395(SQL.105).aspx>).

**Phase 2: Reconfigure Automatic Start Action**

In this phase, you perform steps that reconfigure the automatic start action for the new virtual machine so it is controlled through the cluster service instead of Hyper-V Virtual Machine Management service. The steps you need to follow appear below.

1. Under the Virtual Machines section in Hyper-V Manager, click on the virtual machine created in the previous steps and click Settings under the virtual machine name in the Action pane.
2. Click Automatic Start Action in the Management section in the left pane of the Settings dialog box.
3. Set the Automatic Start Action option to Nothing under What do you want this virtual machine to do when the physical computer starts?
4. Click Apply.

**Phase 3: Configure the Virtual Machine for High Availability**

The final step in creating a virtual machine with the Hyper-V Manager console is making it highly available in Failover Cluster Manager. As a result, the virtual machine is able to take advantage of Live Migration. Here are the steps you need to perform to achieve high availability:

1. Launch the Failover Cluster Manager Console and connect to the Hyper-V failover cluster managing the virtual machine. If you do not see the cluster in the console tree then right-click Failover Cluster Manager. Click Manage a Cluster and then select or specify the cluster.
2. Next, expand the tree under the cluster you want to configure if it is collapsed and then click Services and Applications.
3. In the Action pane, click Configure a Service or Application.
4. On the Select Service or Application page, select Virtual Machine from the list of options. Click Next.
5. On the Select Virtual Machine page, select the virtual machine you want to configure for high availability and then click Next.
6. Confirm the information on the Confirmation page and then click Next.
7. On the Summary page, review the configuration details by clicking on View Report. Click Finish to close the wizard.
8. Look in one of two places in the console tree to confirm you now have a highly available virtual machine. You can expand Services and Applications in the console tree. The virtual machine should appear under Services and Applications. Alternatively, you can expand Nodes. Find the node the virtual machine was created on and select it in the Results pane under Services and Applications.
9. The virtual machine is ready to be brought online and started. Select the virtual machine under Services and Applications and then click Start Virtual Machine.
10. Now, turn your attention to the steps in section 3.3 to initiate Live Migration for a SQL Server VM.

### 3.2.2. Creating a SQL Server Virtual Machine with Virtual Machine Manager R2

As mentioned earlier, you can also create a virtual machine using Virtual Machine Manager (VMM) R2. Before creating a VM with Virtual Machine Manager R2, however, you need to ensure the failover cluster hosts and affiliated nodes are added to Virtual Machine Manager R2. This will allow the newly created virtual machine to be automatically placed and configured for high availability on the Hyper-V failover cluster. You can accomplish this task using the Add Hosts Wizard in the Virtual Machine Manager 2008 R2 Administrator Console. Once the failover cluster host is added to Virtual Machine Manager 2008 R2, perform the following actions to create a highly available SQL Server virtual machine.

1. Launch the Virtual Machine Administrator Console by selecting Start, All Programs, Microsoft System Center, Virtual Machine Manager 2008 R2, and then Virtual Machine Manager Administrator Console.
2. On the Actions pane, select New Virtual Machine.
3. On the Select Source page, select the option Create the New Virtual Machine with a Blank Virtual Hard Disk and then click Next.
4. On the Virtual Machine Identify page, specify the name of the SQL Server virtual machine, the owner. A description is not required, but you may include one. Click Next to continue.
5. On the Configure Hardware page (see Figure 15), configure the hardware, priority and availability settings for the SQL Server Virtual Machine and then click Next. You will need to include information about the CPU, Memory, Floppy Drive, DVD or CD Drive, Virtual Hard Disk, and Network Adapters based on your SQL Server guest operating system requirements. If you plan on using Live Migration, ensure you select the Availability setting Make this VM Highly Available. Click Next to continue. If you have already created a hardware profile, then you have the opportunity to import the settings by selecting them in the Hardware Profile drop-down list. For example, if you are provisioning many SQL Server virtual machines based on similar hardware attributes or you want to maintain a hardware standard, it is beneficial to create a hardware profile that can be used for all virtual machines.

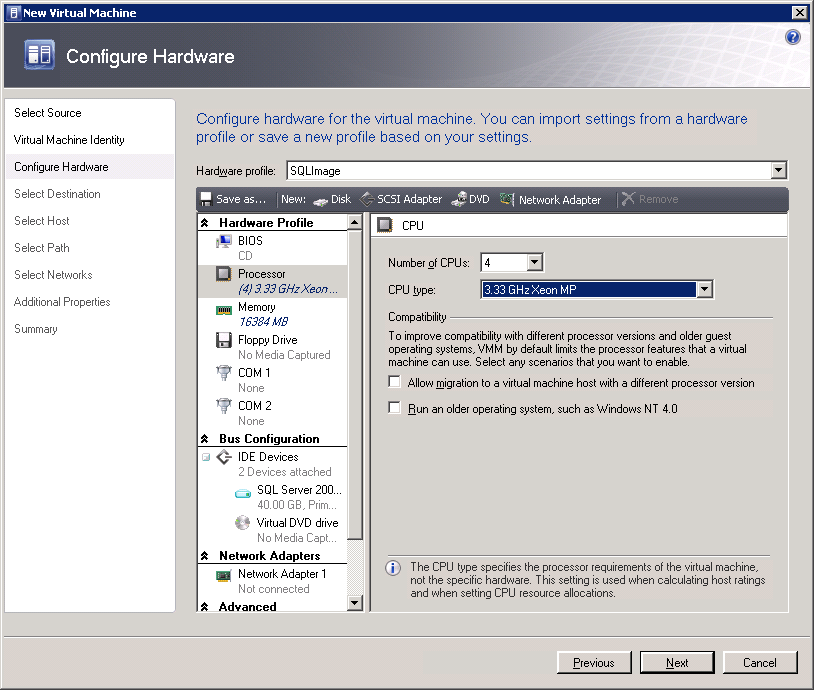


Figure 15. Configuring Hardware for a new SQL Server Virtual Machine.

1. On the Select Destination page, you have the opportunity to place the virtual machine on a host within the Failover Cluster or you can store the virtual machine in the library for future deployment or rapid deployment of sys-prepped SQL Server images. For the purpose of this guide, place the virtual machine on a host and then click Next.
2. On the Select Host page, specify the Hyper-V physical server that is going to host the SQL Server virtual machine and then click Next. Hyper-V Hosts are rated on the virtual machines’ hardware and resource requirements and the resource consumption on each physical host. Each host has a 5 star rating with 5 stars being the best and 1 star means being the worst.
3. On the Select Path page, specify the path to the virtual machine and then click Next. Remember, the path must reside on a Cluster Shared Volume.
4. On the Networking page, connect the network adapter to an existing virtual network by selecting the appropriate network adapter from the drop down menu and then click Next.
5. On the Additional Properties page, specify the automatic start and stop options and specify the Operating system that will be installed on the virtual machine. Click Next to continue.
6. On the Summary page, configure the setting to start the virtual machine after deploying it on the host. Review the setting and then click Create.
7. Install the Windows Server operating system.
8. Install SQL Server.

## 3.3. Initiating Live Migration for a SQL Server VM

Independent of the method you choose to create a SQL Server virtual machine, Hyper-V Manager or the Virtual Machine Manager R2 Administrator Console, the next steps remain the same. The next steps demonstrate how to use Live Migration to move a running SQL Server virtual machine from one Hyper-V physical host to another without a disruption in service or perceived downtime.

**Initiating Live Migration via the Hyper-V Manager Console**

1. In the Failover Cluster Manager snap-in, if the cluster that will be configured is not displayed in the console tree, right-click Failover Cluster Manager.
2. Next, click Manage a Cluster, and then select or specify the cluster. Expand Nodes.
3. In the console tree located on the left side, select the node of the virtual machine hosting the VM that will be moved to using Live Migration.
4. Right-click the virtual machine resource displayed in the center pane and then click Live Migrate Virtual Machine To Another Node.
5. Select the node the virtual machine will be moved to in the migration.
6. Verify the virtual machine successfully migrated over to the node selected by looking for the VM under the new node in Current Owner.

**Initiating Live Migration via Virtual Machine Manager R2**

1. Launch the Virtual Machine Administrator Console by selecting Start, All Programs, Microsoft System Center, Virtual Machine Manager 2008 R2, and then Virtual Machine Manager Administrator Console.
2. In Virtual Machines view, navigate to the host on which the virtual machine is deployed in the navigation pane.
3. In the results pane, select the virtual machine, and then click Migrate in the Actions pane.
4. On the Select Host wizard page (see Figure 16), select a host to deploy the virtual machine on. Since the Virtual Machine is configured in a Hyper-V failover cluster running on Windows Server 2008 R2, the transfer type will be Live Migration and that is depicted under the Transfer Type Column.

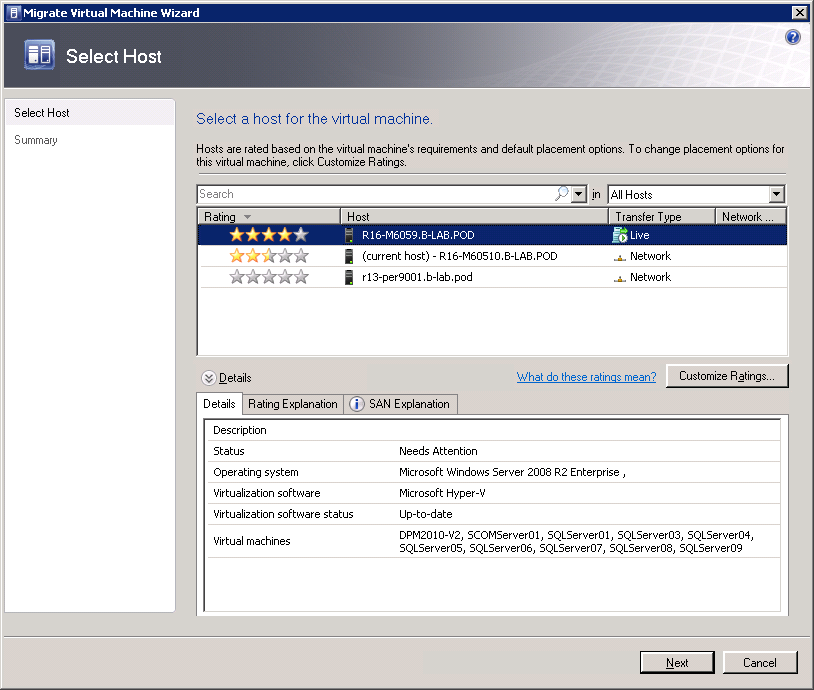


Figure 16. Intelligent Placement of a Virtual Machine on a Hyper-V host.

1. On the Summary wizard page, review your settings and click Move to migrate the virtual machine with Live Migration. Alternatively, you can click Previous to change settings.
2. To begin deploying the virtual machine, click Move.
3. To review the progress and results of the Live Migration operation, display the Jobs window. The window is opened by default when the wizard closes. To view this window at any time, click the Jobs button on the console toolbar.

**Note:** Alternatively, it is possible to enable Performance and Resource Optimization (PRO) in Microsoft System Center Virtual Machine Manager (VMM) 2008 or System Center Virtual Machine Manager 2008 R2 to begin receiving PRO tips in the PRO Tips window of the VMM Administrator Console. Each PRO tip can recommend or automatically implement actions such as virtual machine migration or virtual machine right-sizing based on policies implemented through Operations Manager 2007. For more information, see [About PRO](http://go.microsoft.com/fwlink/?LinkID=157194): (<http://go.microsoft.com/fwlink/?LinkID=157194>).

# 4. Supporting a Virtualized SQL Server Environment

With planning and implementing SQL Server virtual machines with Windows Server 2008 R2 Hyper-V and Live Migration covered, the next part of the whitepaper provides key information on supporting your virtualized SQL Server environment. Discussion on support is broken into five parts:

1. Patch and Service Pack Management
2. Tools to Manage SQL Server Virtual Machines
3. System Center Operations Manager 2007 R2 Integration
4. How to upgrade a SQL Server Instance within Virtual Machine
5. Backing Up SQL Server Virtual Machines

## 4.1. Patch and Service Pack Management

Similar to a physical infrastructure, it is a best practice to patch SQL Server virtual machines regularly. Patching includes applying security updates, critical updates and service packs for the SQL Server and also the guest operating system for each virtual machine that is part of your virtual infrastructure. Moreover, it is equally important to patch the physical Hyper-V host running all of the SQL Server virtual machines. These are the three most common strategies for applying patching:

* Windows Update
* Windows Server Update Services
* System Center Configuration Manager 2007 R2

### 4.1.1. Windows Update

You can use Windows update to apply patches to virtual machines and Hyper-V hosts. This service is built into the Windows Operating System and can be invoked by clicking Start, All Programs, and Windows Update. To find out if updates are needed for Windows and also your programs, hardware or devices, click on Check for Updates in the left pane of the window. Updates for other Microsoft products such as SQL Server can also be obtained by clicking on Change settings. Under the Microsoft Update section enable the setting, *Give me updates for Microsoft products and check for new optional Microsoft software when I update Windows*. Once the investigation for new updates is complete, important and recommended updates are reported. You can click on Install updates or review the updates that are available and select the ones you want to apply by clicking on Install Updates. You can manually run Windows Update on an incremental basis. Alternatively, you can download and install updates on a scheduled basis such as every day at 3:00am.

Windows Update is fairly straightforward to use, however, it can be a tedious task if you plan on manually updating several SQL Server virtual machines, as you will need to run and apply updates on every machine. In addition, there isn't a central console to verify that updates were successfully applied and internet bandwidth is not optimized as each system downloads and stores all of the updates on the local machine. When applying updates to many SQL Server virtual machines, Windows Server Updates Services or System Center Configuration Manager may be a better choice.

### 4.1.2. Windows Server Update Services 3.0 SP2 (WSUS)

Unlike Windows Update, WSUS allows network administrators to fully manage the distribution of Windows and Microsoft updates to one or several servers running on the Windows Operating System in a centralized manner. As a result, administrators can download and deploy updates with little effort to their virtual infrastructures and Hyper-V hosts. For those running Windows Server 2008 R2, begin by adding the Windows Server Update Services server role in Server Manager. Keep in mind when you are installing the role, you will need to provide a location where client computers can obtain updates, specify a database server to house the WSUS repository, and indicate the website that will be used for Windows Server Update Services 3.0 SP2 Web Services. The general practice is for an administrator to launch the console by selecting Start, All Programs, and Windows Server Update Services. Administrators should then move on to downloading updates to the WSUS server. This can be achieved by synchronizing the updates directly from Microsoft's Windows Update. Once WSUS is synchronized with Windows Update, determine, approve and deploy the updates to the target computers within your infrastructure. Target computers include physical machines, virtual machines and Hyper-V hosts. Throughout the process, administrators can use the dashboards and initiative reports to understand what updates are available, what computers are assigned to the WSUS server, and what the deployment status is on each computer being managed. By managing updates in a centralized fashion, an administrator could easily download updates and deploy them to their SQL Server virtual infrastructure including the Hyper-V hosts. For more information on Windows Server Update Services visit the home page here: (<http://technet.microsoft.com/en-us/wsus/default.aspx>).

### 4.1.3. System Center Configuration Manager 2007 R2

Another means for updating your SQL Server virtual machines is with System Center Configuration Manager 2007 R2. System Center Configuration Manager 2007 R2 assesses, deploys, and updates servers, client computers, and devices across physical, virtual, distributed, and mobile environments. A software update feature is also incorporated into the product that can be used to centrally manage and deploy updates to SQL Server virtual machines. The software update management capability is much richer than WSUS as it can deliver updates of Microsoft products, third-party applications, custom in-house line-of-business applications, hardware drivers, and system BIOS to a variety of devices including desktops, laptops, servers, virtual machines and mobile devices. Not to mention that the product differs from WSUS in view of the fact that software updates are only one of its capabilities. Specifically, System Center Configuration Manager 2007 R2 is also used for asset intelligence, configuration management, software distribution and operating system deployment. Another difference is that unlike WSUS, this product is not a free download or included with Windows Server 2008 R2; it is a separate Microsoft product in the System Center suite of products that must be purchased.

### 4.1.4. Applying Updates and Service Packs

Independent of the method you use to patch your virtual machines, it is still necessary to apply updates to the guest operating system, the SQL Server application running in the virtual machine, and the Hyper-V hosts running the SQL Server virtual machines. The process of applying updates and service packs to SQL Server virtual machines is the same as when working with physical machines. As such, expect to reboot the SQL Server virtual machine when applying updates or services packs. However, you can use SQL Server high availability alternatives such as failover cluster, database mirroring and peer-to-peer replication in conjunction with a SQL Server virtual machine in order to reduce downtime during updates. For example if you leverage guest clustering it is possible to apply Windows and SQL Server updates including service packs in a rolling upgrade fashion to minimize SQL Server downtime. To assist you, here is the general rolling upgrade process for SQL Server guest clusters:

1. Install Windows and SQL Server updates or service packs on the SQL Server passive node(s).
2. Restart passive node(s) as prompted.
3. Failover SQL Server instance from active node to passive node.
4. Install Windows and SQL Server updates or service packs on the SQL Server active node which is now a passive node as you failed over the services in step 3.
5. Restart the node in step 4.

For more information on SQL Server 2008 failover cluster rolling patch and service pack process review the following link: (<http://support.microsoft.com/kb/958734>).

**Note:** A rolling upgrade strategy can also be applied if you plan on updating each Hyper-V physical host in the failover cluster. The steps include:

1. Install Windows updates or service packs on the Hyper-V passive physical node(s).
2. Restart passive node(s) as prompted.
3. Use Live Migration to failover SQL Server virtual machines from active node to passive nodes in the Hyper-V failover cluster.
4. Install Windows updates or service packs on the Hyper-V active node which is now a passive node as you failed over the services in step 3.
5. Restart the node in step 4.

## 4.2. Tools to Manage SQL Server Virtual Machines

There are a number of tools included in Virtual Machine Manager 2008 R2 and SQL Server 2008 R2 to help you manage your SQL Server virtual machine environment. The upcoming sections explore the different tools and discuss how best to leverage them.

### 4.2.1. The VMM 2008 R2 Administration Console's Virtual Machine View

A large number of operations can be performed from the Virtual Machine view in the Virtual Machine Manager 2008 R2 Administration Console including creating, deploying, migrating, operating, cloning, repairing, storing, and removing virtual machines. In short, the Virtual Machine view is used to administer all of the functionality associated with your SQL Server virtual machines. The Results pane, however, can also help you gain quick insight into the status of each virtual machine. Take for example the Results pane below in Figure 17. The first column lists the names of the virtual machines while the second column indicates the status of each virtual machine. The next three columns all convey equally valuable information including the percentage completed for a job that is running, the physical host running the virtual machine, and the owner of the virtual machine. In the final column the CPU average for each virtual machine is displayed.

You can also click on each virtual machine to access three tabs in the lower left screen of the details pane. Click on the Summary tab, Networking and Storage tab, or Latest Job tab for additional information on each SQL Server virtual machine. Finally, leverage the Filters section in the left hand pane to customize the information that appears in the results pane. Based on Figure 17, you can see that some of the SQL Server virtual machines are running, stopped and or paused.

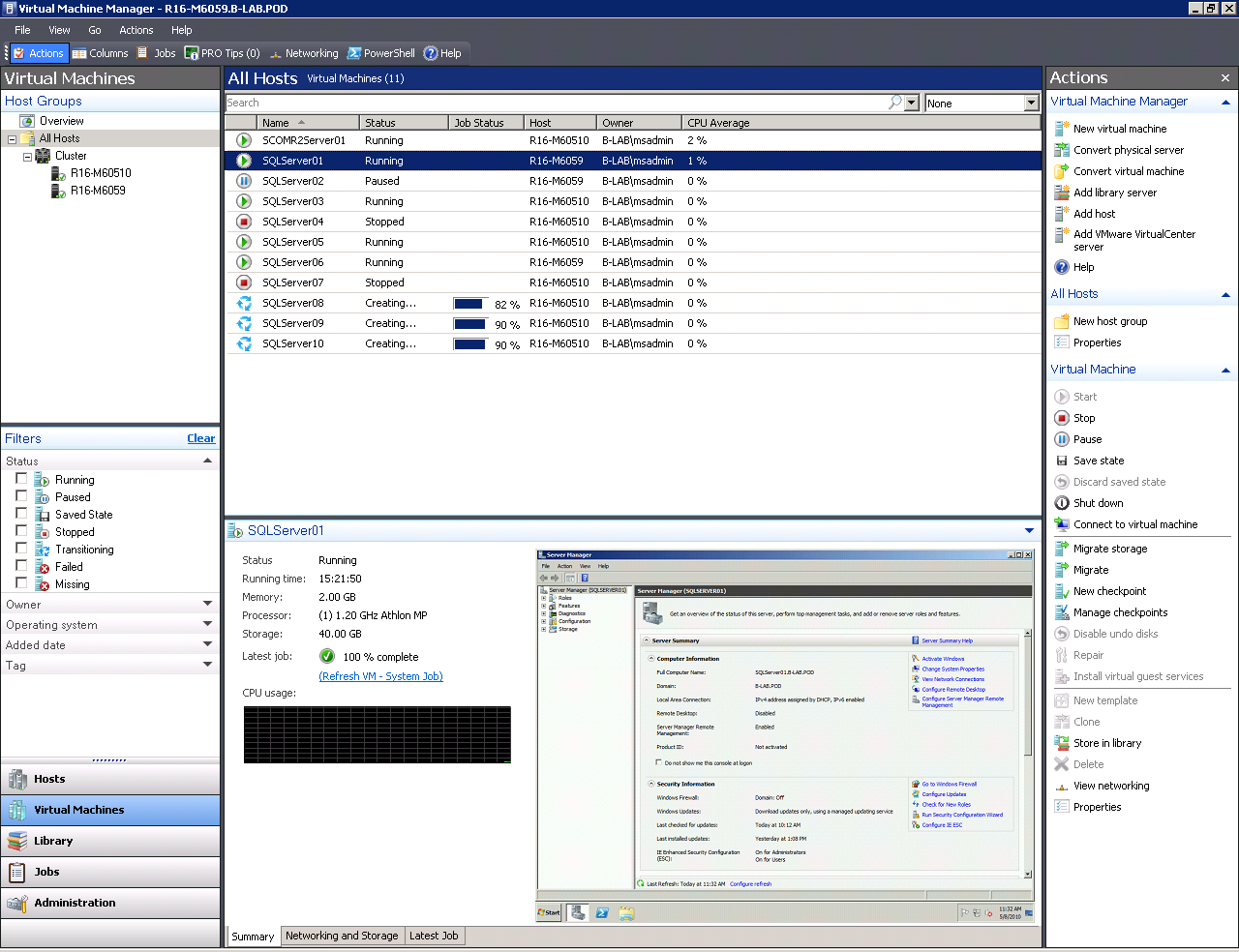


Figure 17. Virtual Machine Manager Administrator Console.

### 4.2.2. The VMM 2008 R2 Administration Console's Administration View

The Administration View in the Administrator Console of Virtual Machine Manager 2008 R2 provides an overview dashboard that communicates the health of the Hyper-V hosts, SQL Server virtual machines, recent jobs and library resources. As revealed in Figure 18, health information is represented visually in pie charts and bar graphs. Use this dashboard including the charts and graphs to easily identify issues with your SQL Server virtual machines such as a physical host that may require attention, a virtual machine that is not running, recent job failures, and resources associated with the library.

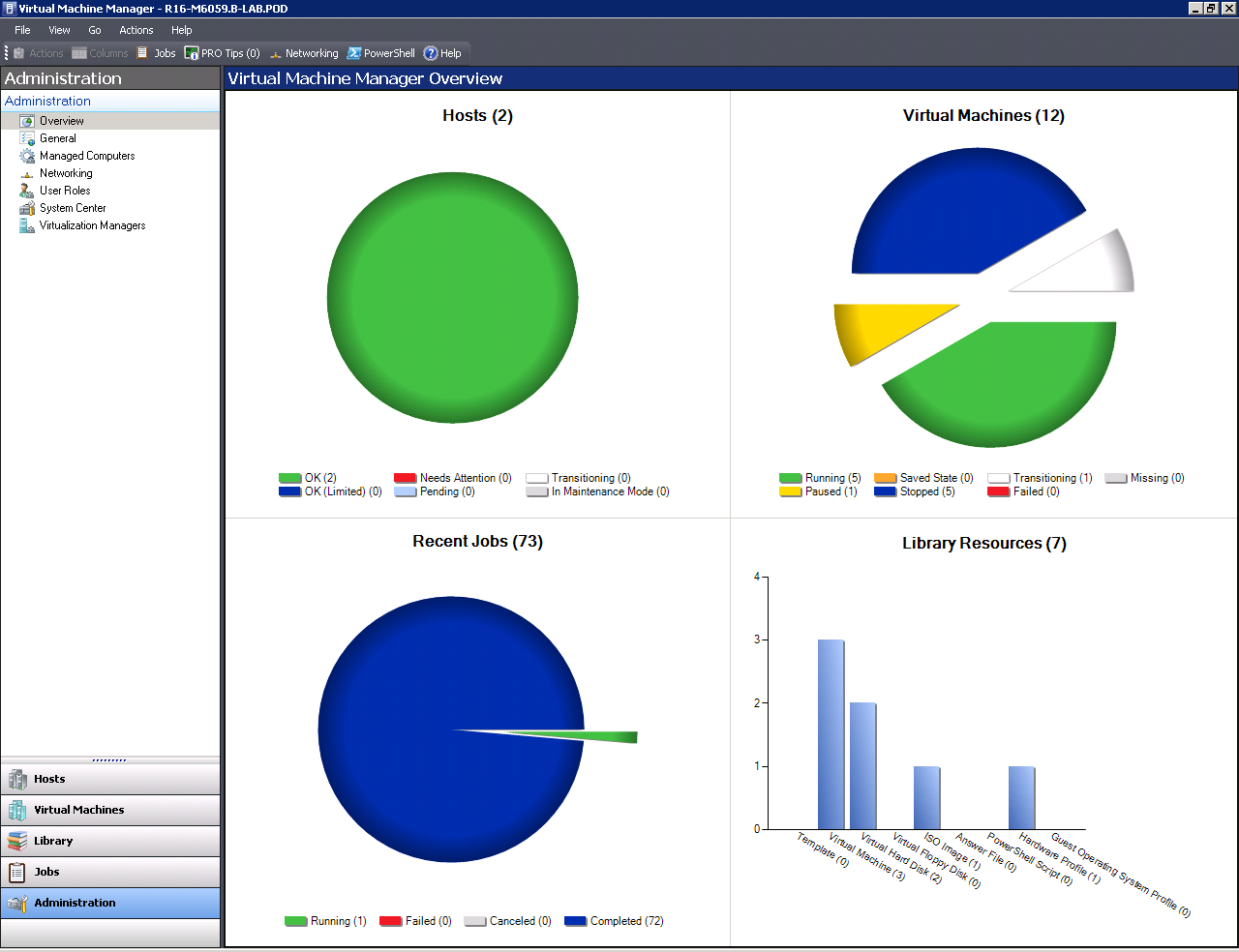


Figure 18. Virtual Machine Manager Overview Screen

### 4.2.3. Utility Control Point

Introduced with SQL Server 2008 R2 is a new capability referred to as the Utility Control Point. It is used to proactively manage database environments at scale through centralized visibility into resource utilization by enrolling SQL Server instances (both physical and virtual) into an instance of SQL Server that serves as the SQL Server Utility. By leveraging the new Utility Explorer and SQL Server Utility dashboards and viewpoints an administrator can monitor SQL Server resource utilization for items such as instance CPU utilization, database files utilization, storage volumes utilization and computer CPU utilization. The Utility Control Point is also used to spot consolidation opportunities by reviewing utilization and policy violation to identify underutilized resources. If you identify that the processor associated with a SQL Server virtual machine or physical host is over-utilized, then you can take advantage of migrating the virtual machine with Live Migration to another host that is under-utilized.

### 4.2.4. Data Collector

The Performance Studio -- Data Collector feature was introduced with the release of SQL Server 2008. It focuses on performance monitoring and troubleshooting SQL Server instances by integrating the collection, analysis, and persistence of SQL Server diagnostics information. A centralized database known as the Management Data Warehouse (MDW) is used to store performance data such as performance counters and dynamic management views from instances of SQL Server across the enterprise. Data collections can include either physical SQL Server instances or virtual SQL Server instances. With the data collected, database administrators can use SQL Server Management Studio to view performance data with intuitive built-in reports that illustrate server activity, processor statistics, memory consumption, wait statistics, disk usage and query activity for all SQL Server virtual machines within the environment.

## 4.3. System Center Operations Manager 2007 R2 Integration

System Center Virtual Machine Manager (VMM) 2008 R2 uses System Center Operations Manager 2007 R2 to monitor the health and availability of the virtual machines managed by VMM 2008 R2. Operations Manager also monitors the components that make up the VMM 2008 R2 infrastructure like the VMM server, database server, library servers, and self-service Web servers.

To leverage critical reports that will help you manage your SQL Server virtual machines, you must integrate Operations Manager with VMM 2008 R2. The installation of Operations Manager 2007 R2 and its integration with Virtual Machine Manager 2008 R2 is above and beyond the scope of a database administrator and this whitepaper. The high level steps for integrating Operations Manager with VMM 2008 R2 are presented below.

1. [Ensure your Operations Manager deployment meets VMM requirements](http://technet.microsoft.com/en-us/library/ee236428.aspx#BKMK_EnsurethatyourOperationsManagerDeployment).
2. [Prepare the VMM server for Operations Manager integration](http://technet.microsoft.com/en-us/library/ee236428.aspx#BKMK_PreparetheVMMserver).
3. [Setup Operations Manager root management server for VMM integration](http://technet.microsoft.com/en-us/library/ee236428.aspx#BKMK_SetupyourOperationsManagerrootmanagementserver).
4. [Configure all other management servers for VMM integration](http://technet.microsoft.com/en-us/library/ee236428.aspx#BKMK_ConfigureYourOtherManagementServers).
5. [Complete VMM configuration](http://technet.microsoft.com/en-us/library/ee236428.aspx#BKMK_CompletetheVMMConfiguration).
6. [Install Operations Manager agents on hosts and virtual machines](http://technet.microsoft.com/en-us/library/ee236428.aspx#BKMK_InstallOperaitonsManagerAgents).
7. [Verify a successful Operations Manager integration](http://technet.microsoft.com/en-us/library/ee236428.aspx#BKMK_VerifyASuccessfulIntegration).

Access to more information on these steps is available at: (<http://technet.microsoft.com/en-us/library/ee236428.aspx>). The next sections provide guidance on how to view reports generated when integrating VMM and Operations Manager.

**Note:** Operations Manager also offers a SQL Server Management Pack that provides both proactive and reactive monitoring of a SQL Server environment. The SQL Server Management Pack works with either physical or virtual machines. If you plan on deploying Operations Manager to view virtual machine reports, then it is a best practice to also implement the SQL Server management pack to obtain numerous views that show state, health and performance information as well as alerts generated according to availability, performance, configuration and security.

### 4.3.1. Viewing Virtual Machine Reports with System Center

The virtual machine reports in System Center should be used to better manage and understand the status and utilization of your SQL Server virtual machines. View the reports from either the Operations Manager Administrator Console or Virtual Machine Manager Administrator Console. Five standard reports are available to you independent of the System Center Administrator Console you use. Take a look at what the five reports offer:

* **Host Utilization** - This report shows the number of virtual machines running on each host and the average usage and total maximum values for host processers, memory, and disk space.
* **Host Utilization Growth** - The percentage of change in resource usage as well as the number of virtual machines running on selected hosts during a specific time period is identified in this report.
* **Virtual Machine Allocation** - Information in this report can be used to calculate chargeback to costs centers for virtual machines. To utilize the information more effectively, set up the report by Cost Center grouping to summarize CPU, memory, disk and network usage for virtual machines within cost centers.
* **Virtual Machine Utilization** - Resource utilization by virtual machines is the focus of this report. Information in this report can help you identity overutilized and underutilized machines, which can determine the changes required in your environment.
* **Virtualization Candidates** - Use this report to identify physical computers that are good candidates for conversion to virtual machines.

**Viewing Virtual Machine Reports with VMM Administrator Console**

Here are the steps for viewing reports for your SQL Server virtual machines with the VMM Administrator Console:

1. Launch the VMM Administrator Console by selecting Start, All Programs, Microsoft System Center, Virtual Machine Manager 2008 R2, and then Virtual Machine Manager Administrator Console.
2. In the VMM Administrator Console, click the Reporting button and then click Reports in the navigation pane.
3. Click the report you want to open. Next, locate the Actions pane on the right side of your screen and click on Open report under the Reporting Actions section.
4. In the report dialog box, enter the parameters that are specific to the report and then click View Report. Report parameters typically consist of start date, end date and the Hyper-V Host's FQDN (Fully Qualified Domain Name).
5. To view a description of the report, expand Description at the top of the report. A summary of the parameters used to produce the report are displayed under the description. To hide the parameters, click the View or Hide Parameter Area button on the toolbar.

Once you have a report opened, it is beneficial to understand how to effectively navigate it. Operations Manager's data warehouse is updated each hour, however, there may be situations when you require data that is more current in an open report. In these situations, you are not required to run the report again. Instead, click the Refresh button on the toolbar. Some reports are several pages in length. Use the arrow buttons located on the toolbar or type the page number in the text box and press Enter to maneuver around a multipage report. Likewise, if you need to find a word or phrase in a multipage document use the Find box. All reports can be printed or exported. Save your report in any of the following formats: XML, CSV (Comma -Separated Values), TIFF (graphic), Acrobat (PDF), Web archive, or Excel.

**Viewing Virtual Machine Reports with Operations Manager Administrator Console**

Here are the steps for viewing reports for your SQL Server virtual machines with Operations Manager:

1. Launch the System Center Operations Manager Operations Console by clicking Start, All Programs, System Center Operations Manager 2007 R2, and then click Operations Console.
2. Display the reports for Virtual Machine Manager by clicking the Reporting button in the navigation pane of the Operations Console.
3. Expand the Reporting folder, and then click System Center Virtualization Reports 2008 R2.
4. In the results pane, select a report that is relevant to a Hyper-V host managing your SQL Server virtual machines.
5. In the Actions pane, click Open to open the report or click Schedule to schedule a periodic report.
6. If required, enter the parameters associated with the report and select Run to view the report.

## 4.4. How to upgrade a SQL Server Instance within a Virtual Machine

Let's say you want to upgrade a SQL Server instance running on a virtual machine from SQL Server 2008 to SQL Server 2008 R2. The process for upgrading a SQL Server instance is the same whether you are running a physical machine or a virtual machine. Here is the high level process you follow whether you are running SQL Server from a physical machine or SQL Server within a virtual machine. First it is a best practice to begin an upgrade by first checking SQL Server 2008 R2 compatibility by running Upgrade Advisor, which is included with SQL Server 2008 R2. The Upgrade Advisor is used to analyze installed components from earlier versions of SQL Server and report issues requiring attention prior to your upgrade. This tool is also useful for identifying deprecated features and providing recommendations for tasks that need to be carried out once the upgrade is complete. The next step is reviewing the Pre-Upgrade checklist included with SQL Server 2008 R2 books online. If you have a green light, perform the Database Engine upgrade to SQL Server 2008 R2 within the guest operating system. Alternatively, you can conduct a data migration from a physical SQL Server to a new SQL Server instance running on a virtual machine. Many organizations take advantage of this strategy if they plan on consolidating several databases running on older versions of SQL Server to the latest data platform such as SQL Server 2008 R2 or if they plan on using pass-through disks during the physical to virtual conversion.

## 4.5. Backing Up SQL Server Virtual Machines

The same amount of attention should be given to backing up the data on SQL Server virtual machines as it is for backing up the data on physical machines. For one reason, the plan of action for backing up SQL Server databases within a virtual machine is the same as backing up SQL Server databases on a physical server. Accordingly, if you took advantage of the native backup tools included in SQL Server Management Studio to back up the system and user databases for each SQL Server instance on a physical machine then use this strategy for backing up your databases in a virtual environment. In addition, you can continue to leverage database snapshots included with SQL Server if you intend to take point-in-time snapshots of your database(s).

Independent of backing up each individual database, serious thought should also be given to backing up the whole SQL Server virtual machine including the Cluster Shared Volumes associated with your SQL Server virtual machines residing on the Hyper-V cluster. To conduct this task, use Data Protection Manager 2010.

### 4.5.1. Using Data Protection Manager (DPM) 2010 to Protect SQL Server Virtual Machines

Data Protection Manager (DPM) 2010 is a member of the System Center family of management products. It delivers continuous data protection for Windows servers and applications like SQL Server and Hyper-V. DPM 2010 also includes new capabilities to protect Live Migration-enabled virtual machines running on CSV in Hyper-V R2. Therefore, DPM 2010 can be used to protect the virtual machines running SQL Server, or it can provide granular database protection. Because DPM's data protection capabilities are far-reaching -- protecting the Live Migration enabled SQL Server virtual machines running on CSV in Hyper-V R2 and protecting the actual databases -- administrators can recover a SQL Server virtual machine and a database to any transaction point, or to the very latest committed transaction following an outage.

In this whitepaper you will learn how to protect all of the Live Migration enabled SQL Server virtual machines within a Hyper-V R2 host running on Cluster Shared Volumes. See the following article for more information on protecting SQL Server databases with Data Protection Manager 2010: (<http://www.microsoft.com/systemcenter/en/us/data-protection-manager/dpm-protect-sql.aspx>).

Before you take steps to protect your SQL Server Virtual Machines, remember to add disks to the DPM storage pool and install a protection agent on each Hyper-V host. If your protection plans include the SQL Server databases then you will also want to install a protection agent on each of the SQL Server virtual machines you plan on protecting.

Protect the SQL Server virtual machines residing on a Hyper-V host with Microsoft System Center Data Protection Manager 2010 by following the steps below:

1. Open the DPM Administrator Console by selecting Start, All Programs, Microsoft System Center Data Protection Manager 2010 and click Microsoft System Center Data Protection Manager 2010.
2. Locate the heading, Actions on the menu bar located at the far right of the screen. Next, click View Protection and then select Create Protection Group under the Selected Item heading.
3. The Create New Protection Group Wizard is invoked. Review the welcome message and then click Next. On the Select Protection Group Type page, select the Servers option to back up file servers and application servers such as Hyper-V and SQL Server. Click Next.
4. The Select Group Members page opens (see Figure 19). Under Available members, expand the Hyper-V server to view the SQL Server virtual machines hosted on the server. Select the SQL Server virtual machines you want to add to the protection group and then click Next.

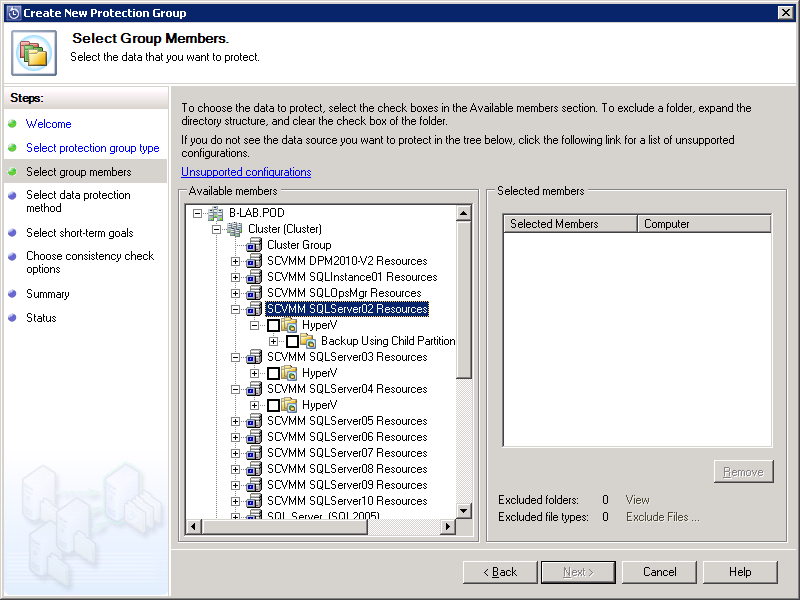


Figure 19. Protecting SQL Server Virtual Machines with DPM 2010.

1. On the Select Data Protection Method page in the text box next to the Protection group name, specify a name for the protection group or accept the default name. Select I want short-term protection using Disk. Click Next to Continue.
2. Use the Specify Short-Term Goals page (see Figure 20) to specify your short-term recovery goals for disk based protection by first indicating in days the Retention Range. Next, either select Every and then choose a value in minutes or hours for Synchronization frequency or select Just before a recovery point. Finally, specify recovery points for file members and then click Next.

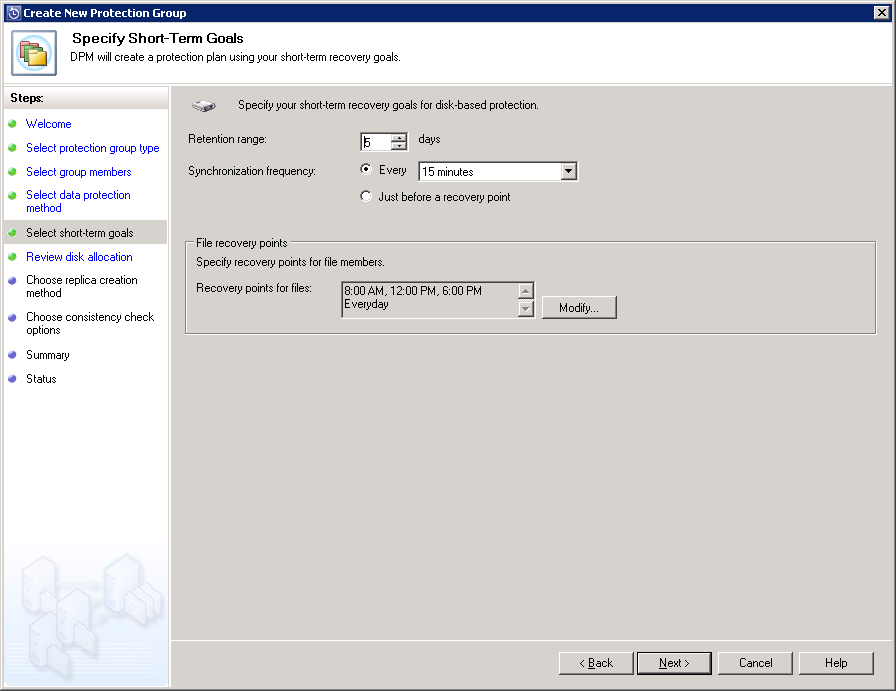


Figure 20. Creating the Protection Plan using your Short-Term Goals.

1. On the Consistency check options page select Run a consistency check if a replica becomes inconsistent. Click Next to move to the next page.
2. On the Review Disk Allocation page (see Figure 21), accept the disk space allocation recommendations or adjust the allocations by clicking Modify. Ensure Automatically Grow the Volumes is selected and then click Next to continue.

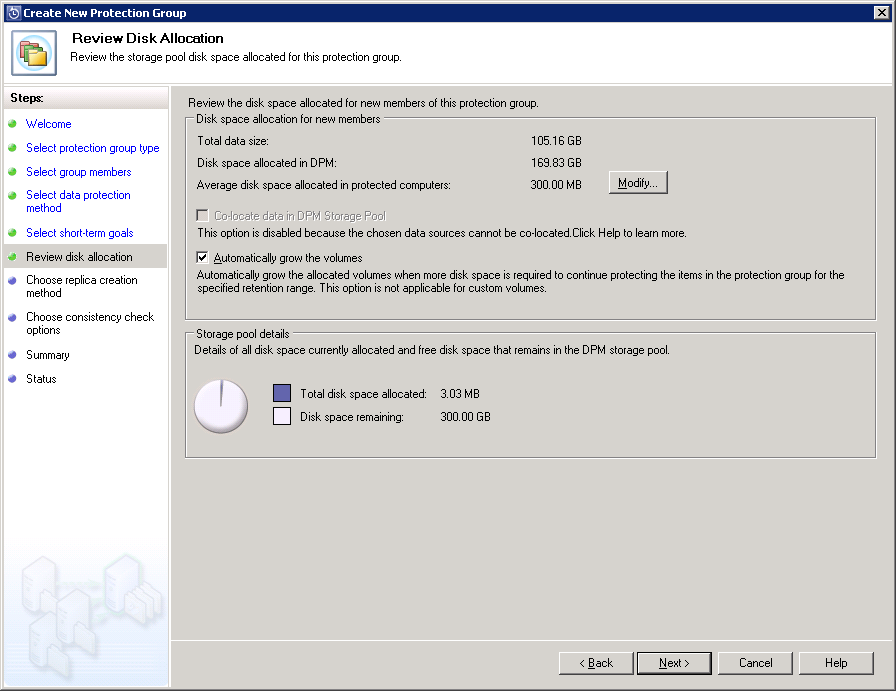


Figure 21. Reviewing the Storage Pool Disk Space Allocation.

1. Once all the protection options are configured, you are given several choices. You can have DPM replicate data at once by selecting Now. Alternatively, you can select Later. If you select Schedule, DPM replicates data based on a predetermined schedule. After you have made a selection, click Next.
2. On the Consistency Check Options page, specify how consistency checks on inconsistent replicas will be run. Click Next to continue.
3. The Summary page appears for your review. Confirm all the settings are correct and then click Create Group.
4. On the Status page, ensure all tasks have been successfully completed. Click close when finished.

# 5. Conclusion

In order to ensure a successful SQL Server virtualization strategy, it is beneficial to first leverage the steps outlined in the planning section of this whitepaper. Second, by choosing Hyper-V it is possible to significantly lower total cost of ownership and maximize return on investment for your SQL Server virtual environment with the improved hardware capabilities included with Windows Server 2008 R2. SQL Server virtualization should also be used as a consolidation strategy to help combat database server sprawl, realize higher return on investment and reduce hardware cost, energy cost and management overheads. Finally, the management capabilities included in the System Center suite should be utilized to proactively monitor, manage and protect the virtualized SQL Server environment.