



SQL Server on VMware Best Practices Guide

© 2012 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. This product is covered by one or more patents listed at <http://www.vmware.com/download/patents.html>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.
3401 Hillview Ave
Palo Alto, CA 94304
www.vmware.com

Contents

1. Introduction	5
1.1 Purpose.....	5
1.2 Target Audience.....	6
1.3 Scope	6
2. SQL Server Performance on vSphere.....	7
2.1 Single Virtual Machine Performance Relative to Native	8
2.2 Multiple Virtual Machine Performance and Scalability.....	9
3. Migrating SQL Server to vSphere	11
3.1 Understand Database Workloads	11
3.2 Understand Availability and Recovery Requirements.....	12
3.3 Capture Resource Utilization Baseline for Existing Databases	14
3.4 Plan the Migration	15
3.5 Understand Database Consolidation Approaches.....	16
4. VMware vSphere Host Best Practices for SQL Server	19
4.1 General Guidelines	19
4.2 CPU Configuration Guidelines	19
4.3 Memory Configuration Guidelines.....	21
4.4 Storage Configuration Guidelines	24
4.5 Networking Configuration Guidelines	34
5. SQL Server In-Guest Best Practices	36
5.1 Maximum Server Memory and Minimum Server Memory	36
5.2 Lock Pages in Memory	36
5.3 Large Pages.....	37
5.4 Using Virus Scanners on SQL Server	38
6. Ongoing Performance Monitoring and Tuning.....	39
6.1 Performance Monitoring Tools.....	39
6.2 Key Performance Metrics on vSphere	43
7. vSphere Enhancements for Deployment and Operations.....	44
7.1 vSphere vMotion, DRS, and vSphere HA.....	44
7.2 Templates	45
7.3 vFabric Data Director	45
7.1 vCenter Operations Management Suite	46
7.2 vCenter Site Recovery Manager.....	48

List of Figures

Figure 1. Scale-Up Performance in vSphere 4 Compared with Native.....	8
Figure 2. Consolidation of Multiple SQL Server Virtual Machines	9
Figure 3. Overcommit Fairness for Eight Virtual Machines	10
Figure 4. Traditional Physical Database Consolidation	16
Figure 5. Database Consolidation on vSphere	17
Figure 6. Scale-Up versus Scale-Out Approach	18
Figure 7. Virtual Machine Memory Settings	22
Figure 8. VMware Storage Virtualization Stack	25
Figure 9. Random Mixed (50% read/50% write) I/O Operations per Second (higher is better)	26
Figure 10. Sequential Read I/O Operations per Second (higher is better)	27
Figure 11. Read Throughput for Different I/O Block Sizes	29
Figure 12. Write Throughput for Different I/O Block Sizes.....	29
Figure 13. Storage Multipathing Requirements for vSphere.....	30
Figure 14. Configure EagerZeroedThick Disk in vSphere 4.x	32
Figure 15. Configure EagerZeroedThick Disk in vSphere 5.x	32
Figure 16. Optimize with Device Separation.....	33
Figure 17. Virtual Networking Concepts	34
Figure 18. Performance Chart Viewed with the vSphere Client	39
Figure 19. resxtp CPU Metrics	40
Figure 20. vFabric Data Director	46
Figure 21. vCenter Operations Manager	47
Figure 22. VMware vCenter Site Recovery Manager	48

List of Tables

Table 1. Scale-Up Performance.....	8
Table 2. Aggregate System Metrics for Eight SQL Server Virtual Machine	10
Table 3. SQL Server 2012 High Availability Options	13
Table 4. Metrics.....	14
Table 5. Overhead Memory Required for Power-On	22
Table 6. VMFS and Raw Disk Mapping Trade-Offs.....	28
Table 7. Typical SQL Server Disk Access Patterns.....	32
Table 8. Key SQL Counters	41
Table 9. Key Performance Metrics.....	43

1. Introduction

Microsoft SQL Server is one of the most widely deployed database platforms in the world, with many organizations having dozens or even hundreds of instances deployed in their environments. The flexibility of SQL Server, with its rich application capabilities combined with the low costs of x86 computing, has led to a wide variety of SQL Server installations ranging from large data warehouses to small, highly specialized departmental and application databases. The flexibility at the database layer translates directly into application flexibility, giving end users more useful application features and ultimately improving productivity.

Application flexibility often comes at a cost to operations. As the number of applications in the enterprise continues to grow, an increasing number of SQL Server installations are brought under life-cycle management. Each application has its own set of requirements for the database layer, resulting in multiple versions, patch levels, and maintenance processes. For this reason, many application owners insist on having an SQL Server installation dedicated to an application. As application workloads vary greatly, many SQL Server installations are allocated more hardware than they need, while others are starved for compute resources.

The challenge for the administrator is to provide database services to application owners with the flexibility and autonomy they expect while keeping the infrastructure as simple and economical as possible. The proliferation of large, multsocket, multicore servers has led many organizations to attempt traditional database consolidation, moving small databases into large shared database environments. Migrating to such a model can be an extremely complex endeavor requiring in-depth application remediation at the forefront and rigorous attention to operational processes after implementation for version control and continued application compatibility.

Virtualizing Microsoft SQL Server with VMware® vSphere® can allow the best of both worlds, simultaneously optimizing compute resources through server consolidation and maintaining application flexibility through role isolation. SQL Server workloads can be migrated in their current states without expensive and error-prone application remediation, and without changing operating system or application versions or patch levels. For high performance databases, VMware and partners have demonstrated the capabilities of vSphere to run the most challenging SQL Server workloads. For smaller, specialized databases, vSphere offers high consolidation ratios and advanced resource scheduling features, giving application owners the flexibility and performance they need while simplifying and lowering costs for the enterprise.

In addition, SQL Server virtual machines are much easier to manage than physical servers. For example, VMware vSphere vMotion® can help to reduce the impact of business or infrastructure changes by migrating live virtual machines to another physical server in case of hardware changes or upgrades, without interrupting the users or their applications. VMware vSphere Distributed Resource Scheduler™ (DRS) can be used to dynamically balance SQL Server workloads and VMware vSphere High Availability (HA) and VMware vSphere Fault Tolerance (FT) can provide simple and reliable protection for SQL Server virtual machines. vSphere is the key to satisfying your organization's need for a rich application environment reliant on flexible database services while simultaneously providing substantial cost savings and unprecedented management capabilities.

1.1 Purpose

This guide provides best practice guidelines for deploying Microsoft SQL Server on vSphere. The recommendations in this guide are not specific to any particular set of hardware or to the size and scope of any particular SQL Server implementation. The examples and considerations in this document provide guidance only and do not represent strict design requirements, as varying application requirements would result in many valid configuration possibilities.

1.2 Target Audience

This guide assumes a basic knowledge and understanding of VMware vSphere and SQL Server.

- Architectural staff can use this document to gain an understanding of how the system will work as a whole as they design and implement various components.
- Engineers and administrators can use this document as a catalog of technical capabilities.
- DBA staff can use this document to gain an understanding of how SQL might fit into a virtual infrastructure.
- Management staff and process owners can use this document to help model business processes to take advantage of the savings and operational efficiencies achieved with virtualization.

1.3 Scope

The scope of this document is limited to the following topics:

- Section 2, SQL Server Performance on vSphere – This section provides background information on SQL Server performance in a virtual machine. It also provides guidelines for conducting and measuring internal performance tests.
- Section 3, Migrating SQL Server to vSphere – From requirements gathering to performance baselining, this section discusses migration and consolidation of SQL Server instances to the vSphere platform.
- Section 4, VMware vSphere Host Best Practices for SQL Server – This section provides best practice guidelines for properly preparing the vSphere platform to run SQL Server. This section includes guidance in the areas of CPU, memory, storage, and networking.
- Section 5, SQL Server In-Guest Best Practices – This section provides best practice guidelines for configuring SQL Server parameter that could affect performance when running on the vSphere platform.
- Section 6, Ongoing Performance Monitoring and Tuning – This section provides best practice guidelines for configuring SQL Server to run on the vSphere platform.
- Section 7, vSphere Enhancements for Deployment and Operations – This section provides a brief look at vSphere features and add-ons that enhance deployment and management of SQL Server.

The following topics are out of scope for this document, but may be addressed in other documentation in this solution kit:

- Availability and Recovery Options – Although this Best Practices Guide briefly covers VMware features that can enhance availability and recovery, a more in-depth discussion of this subject is covered in *Microsoft SQL Server on VMware Availability and Recovery Options*, included in this solution kit.
- Support and Licensing – This information can be found in the *Microsoft SQL Server on VMware: Support and Licensing Guide* included in this solution kit. All information in this section is based on the most current Microsoft material at the time of this writing and is subject to change without notice.

This and other guides in this solution kit are limited in focus to deploying SQL Server on vSphere. SQL Server deployments cover a wide subject area, and SQL Server design principles should always follow Microsoft guidelines.

2. SQL Server Performance on vSphere

VMware vSphere contains numerous performance related features that make it easy to virtualize a resource-heavy database with minimal impact to performance. The improved resource management capabilities in vSphere facilitate more effective consolidation of multiple SQL Server virtual machines on a single host without compromising performance or scalability. Greater consolidation can significantly reduce the cost of physical infrastructure and of licensing SQL Server, even in smaller-scale environments.

In 2009, VMware conducted a detailed performance analysis of Microsoft SQL Server 2008 running on vSphere. The performance test placed a significant load on the CPU, memory, storage, and network subsystems. The results demonstrate efficient and highly scalable performance for an enterprise database workload running on a virtual platform.

To demonstrate the performance and scalability of the vSphere platform, the test performed the following:

- Measured performance of SQL Server 2008 in an 8 virtual CPU (vCPU), 58GB virtual machine using a high-end OLTP workload derived from TPC-E1.
- Scaled the workload, database, and virtual machine resources from 1 vCPU to 8 vCPUs (scale up tests).
- Consolidated multiple 2 vCPU virtual machines from 1 to 8 virtual machines, effectively overcommitting the physical CPUs (scale out tests).
- Quantified the performance gains from some of the key new features in vSphere.

The following metrics were used to quantify performance:

- Single virtual machine OLTP throughput relative to native (physical machine) performance in the same configuration.
- Aggregate throughput in a consolidation environment.

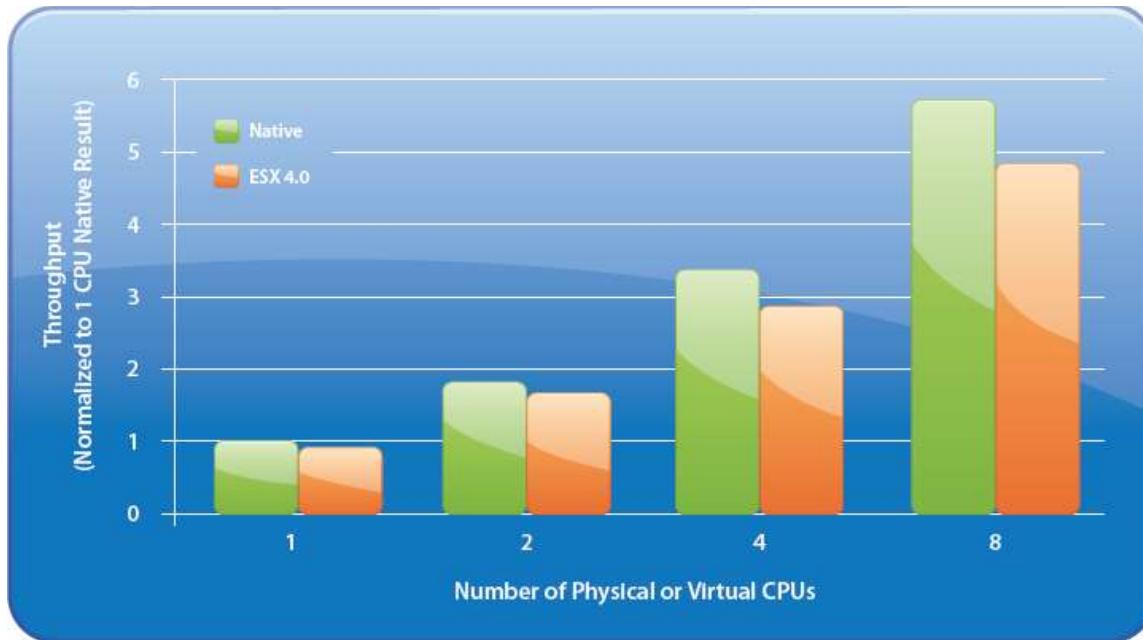
The following sections summarize the performance results of experiments with the Brokerage workload on SQL Server in a native and virtual environment. Single and multiple virtual machine results are examined, and results are given that show improvements due to specific vSphere 4.0 features.

The information in this guide is limited to a summary of test results. See Performance and Scalability of Microsoft SQL Server on VMware vSphere 4 (http://www.vmware.com/files/pdf/perf_vsphere_sql_scalability.pdf).

2.1 Single Virtual Machine Performance Relative to Native

The following figure shows how vSphere 4 performs and scales relative to native. The results are normalized to the throughput observed in a 1 CPU native configuration.

Figure 1. Scale-Up Performance in vSphere 4 Compared with Native



The chart demonstrates the 1 and 2 vCPU virtual machines performing at 92 percent of native. The 4 and 8 vCPU virtual machines achieve 88 and 86 percent of the non-virtual throughput, respectively. At 1, 2, and 4 vCPUs on the 8 CPU server, vSphere is able to effectively offload to idle cores certain tasks such as I/O processing. Having idle processors also gives vSphere resource management more flexibility in making virtual CPU scheduling decisions. However, even with 8 vCPUs on a fully committed system, vSphere still delivers excellent performance relative to the native system.

The scaling in the chart represents the throughput as all aspects of the system are scaled such as number of CPUs, size of the benchmark database, and SQL Server buffer cache memory. The following table shows vSphere scaling comparable to the native configuration's ability to scale performance.

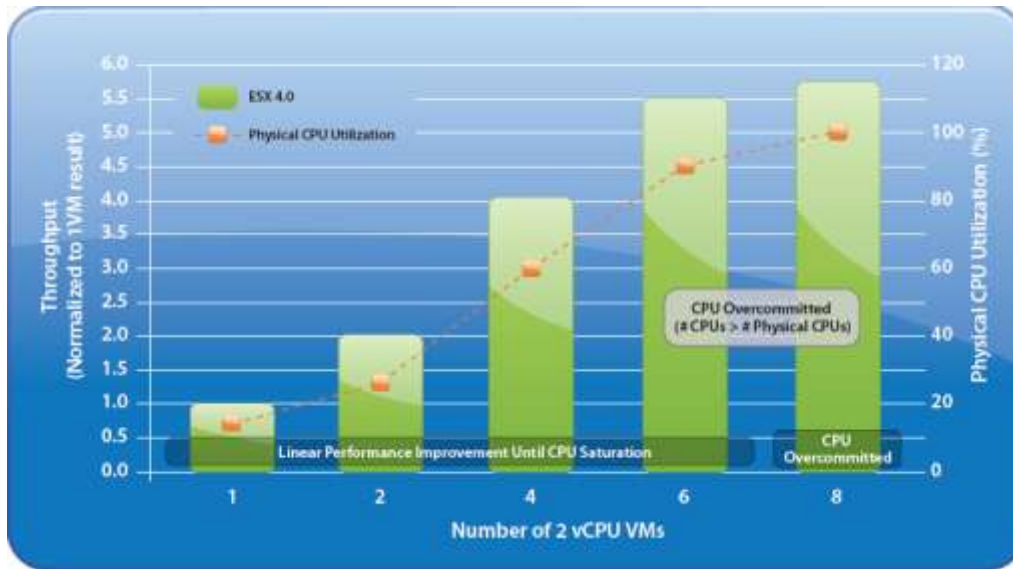
Table 1. Scale-Up Performance

Comparison	Performance Gain
Native 8 CPU versus 4 CPU	1.71
vSphere 8 vCPU versus 4 vCPU	1.67

2.2 Multiple Virtual Machine Performance and Scalability

These experiments demonstrate that multiple heavy SQL Server virtual machines can be consolidated to achieve scalable aggregate throughput with minimal performance impact to individual virtual machines. The following figure shows the total benchmark throughput as eight 2 vCPU SQL Server virtual machines are added to the Brokerage workload onto the single 8-way host.

Figure 2. Consolidation of Multiple SQL Server Virtual Machines



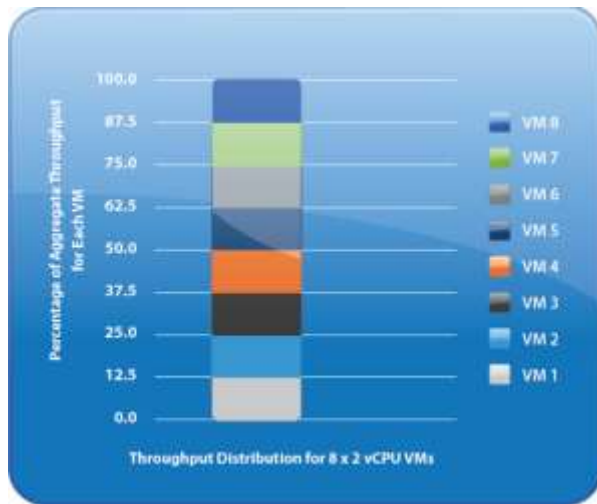
Each 2 vCPU virtual machine consumes about 15 percent of the total physical CPUs, 5GB of memory in the SQL Server buffer cache, and performs about 3600 I/Os per second (IOPS).

As the graph illustrates, the throughput increases linearly as up to four virtual machines (8 vCPUs) are added. As the physical CPUs were overcommitted by increasing the number of virtual machines from four to six (a factor of 1.5), the aggregate throughput increases by a factor of 1.4.

Adding eight virtual machines to this saturates the physical CPUs on this host. vSphere 4.0 now schedules 16 vCPUs onto eight physical CPUs, yet the benchmark aggregate throughput increases a further 5% as the vSphere scheduler is able to deliver more throughput using the few idle cycles left over in the 6 vCPU configuration.

The following figure shows the ability of vSphere to fairly distribute resources in the 8 vCPU configuration.

Figure 3. Overcommit Fairness for Eight Virtual Machines



The following table highlights the resource-intensive nature of the eight virtual machines that were used for the scale out experiments.

Table 2. Aggregate System Metrics for Eight SQL Server Virtual Machine

Aggregate Throughput in Transactions per Second	Host CPU Utilization	Disk I/O Throughput (IOPS)	Network Packet Rate	Network Bandwidth
2760	100%	23K	8Kbps receive 7.5Kbps send	9Mbps receive 98Mbps send

3. Migrating SQL Server to vSphere

When considering SQL Server instances as candidates for virtualization, you need a clear understanding of the business and technical requirements for each instance. These requirements span multiple dimensions, such as availability, performance, scalability, growth and headroom, patching, and backups.

Use the following high-level procedure to simplify the process of identifying SQL Server candidates for virtualization.

1. Understand the database workload requirements for each instance of SQL Server.
2. Understand availability and recovery requirements, including uptime guarantees (“number of nines”) and site recovery.
3. Capture resource utilization baselines for existing databases (on physical).
4. Plan the migration to vSphere.
5. Understand database consolidation approaches.

3.1 Understand Database Workloads

The SQL Server database platform can support a very wide variety of applications. Before deploying SQL Server on vSphere, you must understand the database workload requirements of the applications that your SQL Servers will support. Each application will have different requirements for capacity, performance, and availability, and consequently, each database should be designed to optimally support those requirements. Many organizations classify databases into multiple management tiers, using application requirements to define service level agreements (SLAs). The classification of a database server will often dictate the resources allocated to it.

- Mission critical databases (sometimes referred to as *Tier 1* databases) are considered absolutely essential to your company’s core operations. Mission critical databases and the applications they support often have SLAs that require very high levels of performance and availability. SQL Server virtual machines running mission critical databases might require more careful resource allocation (CPU, memory, disk) to achieve optimal performance. They might also be candidates for database mirroring or failover clustering.
- Other databases and applications are busy only during specific periods for such tasks as reporting, batch jobs, and application integration or ETL workloads. These databases and applications might be essential to your company’s operations, but they have much less stringent requirements for performance and availability. They may, nonetheless, have other very stringent business requirements, such as data validation and audit trails.
- Still other smaller, lightly used databases typically support departmental applications that may not adversely affect your company’s real-time operations if there is an outage. You can tolerate such databases and applications being down for extended periods.

Resource needs for SQL Server deployments are defined in terms of CPU, memory, disk and network I/O, user connections, transaction throughput, query execution efficiency/latencies, and database size. Some customers have established targets for system utilization on hosts running SQL Server, such as 80% CPU utilization, leaving enough headroom for any usage spikes.

Understanding database workloads and how to allocate resources to meet service levels helps you to define appropriate virtual machine configurations for individual SQL Server databases. Because you can consolidate multiple workloads on a single vSphere host, this characterization also helps you to design a vSphere and storage hardware configuration that provides the resources you need to deploy multiple workloads successfully on vSphere.

3.2 Understand Availability and Recovery Requirements

Running Microsoft SQL Server on vSphere offers many options for database availability and disaster recovery utilizing the best features from both VMware and Microsoft. For example, vSphere vMotion and vSphere DRS can help to reduce planned downtime and balance workloads dynamically, and VMware vSphere High Availability (HA) can help to recover SQL Servers in the case of host failure.

3.2.1 vSphere HA, DRS, and vSphere vMotion for High Availability

VMware technologies such as vSphere HA, vSphere DRS, and vMotion can be used in a high availability design.

3.2.1.1. VMware vSphere High Availability

vSphere HA provides easy-to-use, cost-effective, high availability for applications running in virtual machines. In the event of physical server failure, affected virtual machines are automatically restarted on other production servers that have spare capacity. Additionally, if there is an operating system-related failure within a virtual machine, the failure is detected by vSphere HA and the affected virtual machine is restarted on the same physical server.

3.2.1.2. vSphere Distributed Resource Scheduler

vSphere DRS collects resource usage information for all hosts and virtual machines and generates recommendations for virtual machine placement. These recommendations can be applied manually or automatically. DRS can dynamically load balance all virtual machines in the environment by shifting workloads across the entire pool of vSphere hosts so that critical SQL Server virtual machines in the environment always have the CPU and RAM resources needed to maintain optimal performance.

3.2.1.3. vSphere vMotion

vSphere vMotion leverages the complete virtualization of servers, storage, and networking to move a running virtual machine from one physical server to another. This migration is performed with no impact to running workloads or connected users. During vMotion migration, the active memory and execution state of the virtual machine is rapidly transmitted over the network to the new physical server, while maintaining its network identity and connections.

3.2.2 Native SQL Server Capabilities

At the application level, all Microsoft features and techniques are supported on vSphere, including SQL Server AlwaysOn Availability Groups, database mirroring, failover clustering, and log shipping. With the exception of failover clustering, these Microsoft features can be combined with vSphere features to create flexible availability and recovery scenarios, applying the most efficient and appropriate tools for each use case.

The following table lists SQL Server availability options and their ability to meet various recovery time objectives (RTO) and recovery point objectives (RPO). Before choosing any one option, evaluate your own business requirements to determine which scenario best meets your specific needs.

Table 3. SQL Server 2012 High Availability Options

Technology	Granularity	Storage Type	RPO – Data Loss	RTO – Downtime
AlwaysOn Availability Groups	Database	Non-shared	None (with synchronous commit mode)	~3 seconds or Administrator Recovery
AlwaysOn Failover Cluster Instances	Instance	Shared	None	~30 seconds
Database Mirroring	Database	Non-shared	None (with high safety mode)	< 3seconds or Administrator Recovery
Log Shipping	Database	Non-shared	Possible transaction log	Administrator Recovery

For a more in-depth treatment of availability and recovery options for SQL Server, see *SQL Server on VMware Availability and Recovery Options*.

3.2.3 Backup and Restore

The feature set available to an application, when deployed in a virtual environment, is no different than what is available with a physical deployment. In fact, there are more options available for protecting entire virtual machines. This is especially useful for applications that require extensive configuration. For SQL Server, the standard methods for backup are supported. These tend to be deployed using a third-party backup agent that uses a VSS requestor to coordinate with the VSS writer to prepare the database files for backup. Regardless of the backup solution required, VMware and VMware partners have provided solutions for most situations.

To back up and restore SQL Server virtual machines, consider using one of the following options:

- In-guest, software agent-based backup.
- VMware Data Recovery.
- Array-based backup solutions.

Each of these options is covered in *SQL Server on VMware Availability and Recovery Options*.

3.3 Capture Resource Utilization Baseline for Existing Databases

After you clearly understand your organization's needs—business and technical requirements, availability and other operational requirements for implementing SQL Server—the next important step is to establish a baseline, using data from the current, running physical deployment. The best way to accomplish this is to use the Virtualization Assessment service delivered by VMware or its partners. Using VMware Capacity Planner™, this service collects all essential performance metrics, including processor, disk, memory, and network statistics, and specific SQL Server metrics on existing database servers in the environment. Capacity Planner analyzes this data to recommend server consolidation opportunities. You can monitor these metrics, in addition to the essential system metrics, if you want to capture the baseline information yourself. It is important to collect the data over a period of time long enough to reflect all variations in the usage patterns of SQL Server in your organization. The duration can range from one week to one month, depending on seasonal or cyclical usage peaks such as when they occur, their intensity, how long they last, and other factors. This exercise helps you to understand what resources your current physical SQL Servers use, and makes it easier for you to create a short list of SQL Server instances to virtualize and to determine the order in which you should virtualize the SQL Server instances. Refer to the following metrics when running your own analysis. Capture data for average as well as peak periods.

Table 4. Metrics

Resource Type	Perfmon Counter	Note
Processor	Processor(*)\%Processor Time	Percentage of elapsed time the processor spent to execute a non-idle thread.
	Process(sqlservr)\%Processor Time	Percentage of processor time spent executing SQL Server process threads.
	Process(msmdsrv)\%Processor Time	Percentage of processor time spent executing SSAS process threads.
	System\Processor Queue Length	Number of threads waiting in the processor queue.
Memory	Memory\Available Mbytes	Unused physical memory.
	Memory\Pages Input/Sec	Reads from disk per second to resolve hard paging.
	SQL Server:Memory Manager\Target Server Memory (KB)	Amount of memory SQL Server will consume.
	SQL Server:Memory Manager\Total Server Memory (KB)	Amount of memory SQL Server is currently consuming.
	SQL Server:Memory Manager\Memory Grants Pending	Current number of processes waiting for workspace memory grant.
	SQL Server:Buffer Manager\Page Life Expectancy	Number of seconds a page will stay in the buffer pool without references.
Disk	PhysicalDisk(*)\Disk Reads/Sec	Number of read requests issued per second.
	PhysicalDisk(*)\Disk Writes/Sec	Number of write requests issued per second.

Resource Type	Perfmon Counter	Note
	PhysicalDisk(*)\Disk Read Bytes/Sec	Number of bytes transferred per second generated from read requests.
	PhysicalDisk(*)\Disk Write Bytes/Sec	Number of bytes transferred per second from write requests.
	PhysicalDisk(*)\Disk Sec/Read	Average latency, in seconds, for a read of data from disk.
	PhysicalDisk(*)\Disk Sec/Write	Average latency, in seconds, for a write of data to disk.
Network	Network Interface(*)\Bytes Received/sec	Number of bytes received per second.
	Network Interface(*)\Bytes Sent/sec	Number of bytes sent per second.
	Network Interface(*)\Current Bandwidth	Estimated bandwidth of the network interface in bits per second.
	Network Interface(*)\Packets Received/Sec	Number of packets received on the network interface.
	Network Interface(*)\Packets Sent/Sec	Number of packets sent on the network interface.

3.4 Plan the Migration

After you establish baseline profiles for your existing SQL Server databases, the next step is to design a vSphere architecture that meets these profiles. The best practices described in Section 4, VMware vSphere Host Best Practices for SQL Server, can help you to optimize your vSphere environment for SQL Server.

Microsoft offers extensive information on best practices for deploying SQL Server in the Microsoft SQL Server Tech Center (<http://technet.microsoft.com/en-us/sqlserver/default.aspx>) on Microsoft Technet. (Third-party web sites are not controlled by VMware, so links to these Web sites might change.)

These best practices papers provide real-world guidelines and expert tips, and you should follow them for SQL Server deployments on vSphere as well. Similarly, VMware recommends following vendor-specific best practice guidelines for configuring your server hardware, storage subsystems, and network. In general, best practices in physical environments also apply to deployments on VMware vSphere without any changes.

Virtualizing SQL Server should provide benefits that go beyond server consolidation and lower total cost of ownership. A successful SQL Server deployment using vSphere provides better management and administration flexibility as well as higher levels of availability at lower cost.

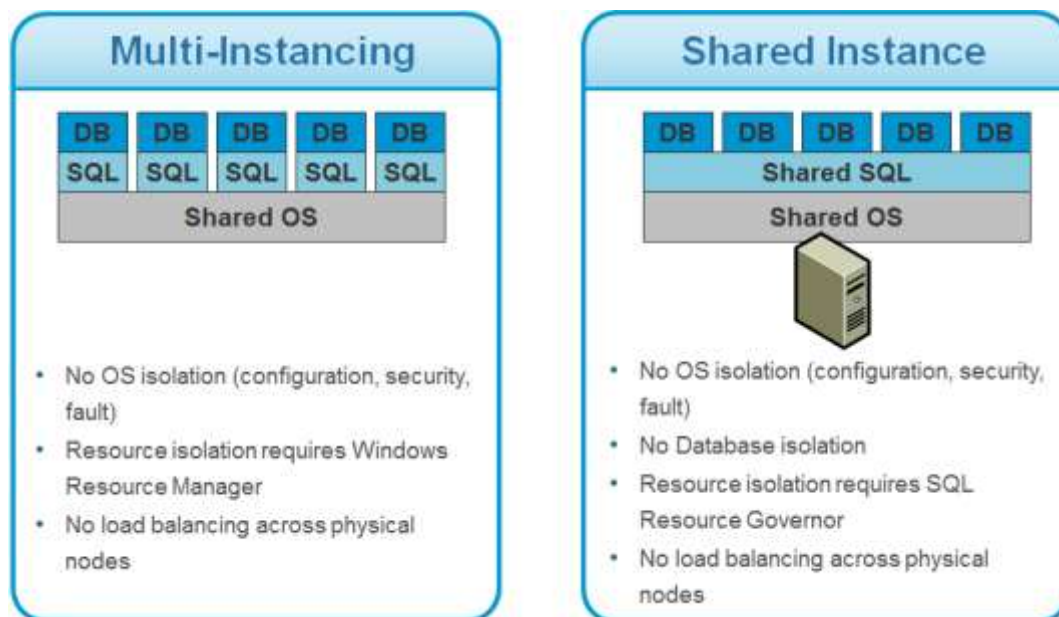
SQL Server databases in an organization have unique deployment needs because of the variety of applications they serve. These applications can require 32-bit or 64-bit versions of the operating system and SQL Server, particular service packs or hot fixes, security and other access control settings, and support for specific legacy application components. When you consolidate such SQL Server databases in physical environments, these specific requirements often force you to take a least common denominator approach to configuration, thus compromising optimal performance. Because a VMware vSphere environment enables you to run both 32-bit and 64-bit virtual machines side-by-side but isolated from each other, virtualized consolidation is much more flexible and much less constrained. However, understanding these specific deployment requirements can help you to refine the virtualization approaches that work best, and can help you to decide whether to adopt a scale-up approach with multiple databases in a single large virtual machine or a scale-out approach with one or only a few databases per virtual machine. You can also identify the optimal workload, security, and other application-specific optimal configurations that need to be locked at the virtual machine level for your databases.

In planning a migration to a virtual environment, you should also understand other deployment and operational requirements, such as those for patching the operating system and SQL Server and those for rolling out and maintaining applications in test and development environments as well as production environments. Understanding these requirements can help you to define how you should create, stage, and provision virtual machines in your datacenter, leveraging VMware vCenter™ Converter™ or the templates and cloning features in vSphere.

3.5 Understand Database Consolidation Approaches

Most legacy database applications were traditionally deployed on dedicated physical hardware. Each database instance provisioned onto its own database server. This approach allows the database administration team to provide predictable service levels to their customer base through isolation of those physical resources. Rogue database instances cannot consume resources from other databases and impact their service levels. A significant limitation of this approach is that each of these databases is bounded by the physical resources available on its physical server and consequently is usually significantly overprovisioned. VMware captured capacity planning data from the servers of the customer base and found that physical database servers use between 5% and 15% of available resources on average, as shown in the following figure. This is a very inefficient use of server hardware, power and cooling resources, as well as database software licenses. Customers have recognized the inefficiency in this approach and in some cases have looked at physical database consolidation as a solution.

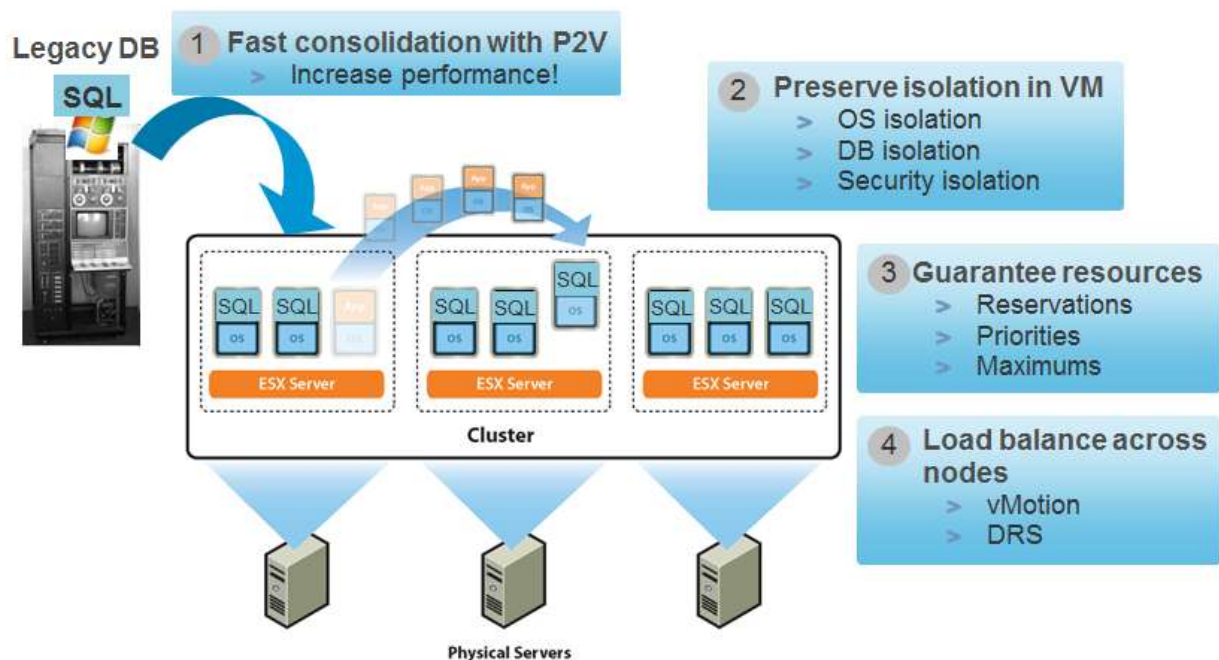
Figure 4. Traditional Physical Database Consolidation



Physical database consolidation involves consolidating multiple SQL Server databases onto a single physical server. This approach is attractive because it enables the consolidation of databases onto fewer physical servers. However, this approach requires making compromises regarding application compatibility, workload isolation, flexibility for maintenance, availability requirements, and security. There are two primary approaches to database consolidation, multi-instance and shared instance consolidation. Both approaches require building a standardized software stack on which to deploy the database instances. Multi-instance database consolidation requires a single standard version of Windows with multiple SQL Server instances, possibly at different version levels, deployed on that single version of Windows. Each instance can have multiple databases. Shared Instance consolidation goes a step further and provides for consolidation on a single SQL Server instance, with multiple databases on that instance.

Both approaches can be challenging to implement. Application owners must determine the compatibility issues associated with their applications, and the Windows and SQL Server versions selected for the consolidation. They must also consider the potential service level implications of both a shared physical infrastructure and a shared software stack. Resources must be governed at the Windows level or within SQL server to limit the potential impact of heavy database activity from one application on the other applications hosted within the consolidated databases. Security models must be re-thought as administrative privilege now provides access to a much broader footprint of applications with new implications. A single bad Windows or SQL Server patch has the potential to impact many more users. These are manageable issues but up-front planning is required to properly address them.

Figure 5. Database Consolidation on vSphere



Database consolidation within a virtual infrastructure provides the benefits of physical database consolidation while also significantly reducing the described implementation challenges. Many customers approach virtual database consolidation by doing a Physical to Virtual (P2V) conversion of each of their physical servers. Because the new virtual machine contains the entire isolated software stack that was on the physical server, there is no reduction in resource isolation from a Windows or SQL Server perspective. There is no need to re-architect the security model within the new Windows guest operating system. vSphere provides the ability to present resources (CPU, memory, and storage) to virtual machines as needed, and to guarantee those resources for applications that require it. These capabilities reduce the need to overprovision the virtual machine to handle peak workloads. The virtual machines also

get increased service levels associated with vSphere High Availability, Fault Tolerance, and vSphere vMotion.

When consolidating databases within a virtual infrastructure, database administrators can adopt a *scale-up* or *scale-out* approach to deploy SQL Server on vSphere.

Figure 6. Scale-Up versus Scale-Out Approach

■ Scale-up approach

- Multiple databases or SQL instances per VM
- Fewer VMs
- Single point of failure
- Larger VM
 - SMP overheads
- OS bottleneck, especially for 32-bit environments



■ Scale-out approach

- Single database per VM
- Better SQL Instance and workload isolation
 - DSS versus OLTP separation
- Easier change management
- DRS more effective with smaller VMs
- Faster migration



The scale-up approach involves consolidating multiple SQL Server instances in a single large virtual machine. The scale-up approach is attractive because it enables you to consolidate onto fewer physical servers and may provide some Windows licensing advantages. However, this approach requires making similar compromises discussed earlier regarding application compatibility, workload isolation, flexibility for maintenance, availability requirements, and security. Furthermore, in a virtual infrastructure, deploying additional virtual machines to house additional instances of SQL Server is relatively painless.

The larger virtual machines also exhibit higher performance overhead associated with the scaling of SMP virtual processors. The scale-up approach may result in bottlenecks in the operating system itself, especially if you reach 32-bit operating system memory limits. Additionally, using larger virtual machines also makes it harder for DRS to move the virtual machine due to the greater demand placed on available resources.

In a virtual infrastructure, these drawbacks to the scale-up approach generally outweigh the benefits. The scale-out approach is generally more appropriate for a virtual infrastructure. With the scale-out approach, you deploy fewer SQL instances per virtual machine and customize the configuration as needed. The usual drawbacks to a scale-out approach that you encounter in a physical infrastructure, such as server sprawl and high TCO, are minimized when you deploy a virtual infrastructure. This approach provides better workload and security isolation, and allows easier maintenance and change management because of the increased granularity of deploying fewer SQL instances per virtual machine. DRS can function more effectively with smaller virtual machines, with the added benefit of faster vSphere vMotion migrations.

4. VMware vSphere Host Best Practices for SQL Server

A properly designed vSphere host platform is crucial to the successful implementation of enterprise applications such as SQL Server. Before we address best practices specific to the SQL Server application, the following sections outline general best practices for designing your vSphere hosts.

4.1 General Guidelines

The following sections discuss general best practice guidelines.

4.1.1 Use VMware vSphere

VMware Server, VMware Workstation™, and even VMware Fusion® are hosted products and are all technically capable of running SQL Server. However, we strongly recommend using a VMware enterprise-class hypervisor, vSphere, to deploy virtualized SQL Server instances, even for development and test environments. When using the hosted products, depending on configurations and guest operating system support, the disk I/O caching performed by the host operating system can provide unpredictable performance and application availability results under specific conditions.

4.1.2 Use the Latest Hardware

x86-based servers and storage systems continue to become more capable than ever, at lower cost. With support for multicore CPUs, larger processor caches, and hundreds of gigabytes of RAM, x86-based servers can now rival mainframe systems. Furthermore, both Intel and AMD have made significant improvements in the performance of hardware-assisted virtualization, allowing virtualized workloads to achieve near-native performance. These hardware advances can significantly alleviate CPU, memory, and disk I/O bottlenecks when consolidating SQL Server virtual machines on a vSphere host.

- Follow SQL Server best practices. Microsoft offers extensive best practices for deploying SQL Server. Best practices are documented on Microsoft TechNet in *SQL Server Best Practices* (<http://technet.microsoft.com/en-us/sqlserver/bb671430.aspx>). These best practices are based on real world guidelines and expert tips, and should be followed for virtual SQL Server deployments as well.
- The vSphere hosts should be sized with adequate capacity to provide resources for all running virtual machines and have enough headroom to account for normal workload variability. This is especially important when various virtual machines exhibit similar workload profiles, and are likely to bottleneck in contention for the same resources.

4.2 CPU Configuration Guidelines

SQL Server virtual machine configuration usually depends on the specific database profile. A thorough virtualization exercise greatly simplifies virtual machine sizing. In general, follow the guidelines discussed in the following sections.

4.2.1 Physical and Virtual CPUs

VMware uses the terms *virtual CPU* (vCPU) and *physical CPU* (pCPU) to distinguish between the processors within the virtual machine and the underlying physical x86/x64-based processor cores. Virtual machines with more than one virtual CPU are also called *symmetric multiprocessing* (SMP) virtual machines. The *virtual machine monitor* (VMM) is responsible for virtualizing the CPUs. When a virtual machine starts running, control transfers to the VMM, which is responsible for virtualizing guest operating system instructions.

4.2.2 Virtual SMP

VMware Virtual Symmetric Multiprocessing (Virtual SMP) enhances virtual machine performance by enabling a single virtual machine to use multiple physical processor cores simultaneously. vSphere supports the use of up to 64 virtual CPUs per virtual machine. Using a large number of vCPUs in a virtual machine can potentially help large-scale, mission-critical, SQL Server applications achieve higher performance and throughput. The ability to use a large number of concurrent threads to execute a T-SQL query can significantly reduce runtime for a complex reporting query.

Since vSphere 5.0 virtual hardware version 8, virtual NUMA (vNUMA) exposes host NUMA topology to the guest operating system, enabling NUMA-aware guest operating systems and applications to make the most efficient use of the underlying hardware's NUMA architecture.

Virtual NUMA, which requires virtual hardware version 8, can provide significant performance benefits for virtualized SQL Servers that feature NUMA optimization.

The VMware Performance Engineering lab conducted a variety of experiments, including commercial Tier 1 and HPC (high-performance computing) applications. Performance was observed to be 92–97% of native, as virtualized applications scaled to 32 vCPUs.

4.2.3 Allocating CPU to SQL Server Virtual Machines

Database server utilization varies widely by application and deployment topology. If the application is commercial, follow published guidelines where appropriate. If the application is custom-written, work with the application developers to determine resource requirements.

4.2.3.1. Properly Manage pCPU Allocation

Tier 1 SQL Server workloads typically require low latency, so maximizing performance is generally the primary goal for these workloads.

Do not oversubscribe the vSphere host. If a host becomes CPU-saturated (the virtual machines and other loads on the host demand all host CPU resources), SQL Server workload performance degrades.

Provide CPU resources by maintaining a 1:1 ratio of the physical cores to vCPUs. For example, if the workloads run on a 4 CPU core machine in the physical system, allocate 4 vCPU on the vSphere host for the virtual machine.

Lower-tier SQL Server workloads typically are less latency sensitive, so in general the goal is to maximize use of system resources rather than performance.

The VMware vSphere CPU scheduler's policy is tuned to balance between maximum throughput and fairness between virtual machines. For lower-tier databases, a reasonable CPU overcommitment can increase overall system throughput, maximize license savings, and continue to maintain adequate performance.

4.2.3.2. NUMA Considerations

VMware vSphere supports AMD (such as Opteron and Barcelona), Intel (such as Nehalem, Westmere), and IBM (X-Architecture) non-uniform memory access (NUMA) systems. The intelligent, adaptive NUMA scheduling and memory placement policies in vSphere can manage all virtual machines transparently, so administrators do not need to deal with the complexity of balancing virtual machines between nodes by hand. To reduce memory access latency, consider the following:

- For small SQL Server virtual machines, allocate virtual machine CPUs equal to or less than the number of cores in each physical NUMA node.
- For wide SQL Server virtual machines, size virtual machine CPUs to align with physical NUMA boundaries. Configure vNUMA to enable SQL Server NUMA optimization to take advantage of managing memory locality.

By default, vSphere NUMA scheduling and related optimizations are enabled only on systems with a total of at least four CPU cores and with at least two CPU cores per NUMA node.

4.2.3.3. Hyperthreading

Hyperthreading technology exposes two hardware contexts from a single core. Hyperthreading doubles the number of available threads on the host. By keeping the processor pipeline busier, hyperthreading generally improves the overall host throughput from 10% to 30%.

Hyperthreading Sharing on the **Properties** tab of a virtual machine provides control of whether a virtual machine should be scheduled to share a physical processor if hyperthreading is enabled on the host. Choose one of the following settings:

- **Any** – This is the default setting. The vCPUs of this virtual machine can freely share cores with other virtual CPUs of this or other virtual machines.
- **None** – The vCPUs of this virtual machine have exclusive use of a processor whenever they are scheduled to the core. Selecting **None** in effect disables hyperthreading for your virtual machine. This option is recommended if you are deploying a Tier 1 SQL Server workload.
- **Internal** – This option is similar to none. Virtual CPUs from this virtual machine cannot share cores with virtual CPUs from other virtual machines. They can share cores with the other virtual CPUs from the same virtual machine.

See additional information on hyperthreading on a vSphere host in *vSphere Resource Management* (<http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-50-resource-management-guide.pdf>).

Enabling hyperthreading can potentially boost overall host throughput. The effects are typically more visible with a highly consolidated SQL Server environment. A processor thread exposed using hyperthreading is not equivalent to a physical CPU thread. If you plan to enable hyperthreading on the virtual machine, remember to account for the differences between a processor thread and a physical CPU thread during capacity planning for your SQL Server deployment.

4.3 Memory Configuration Guidelines

This section provides guidelines for allocation of memory to SQL Server virtual machines. The guidelines outlined here take into account vSphere memory overhead and a virtual machine's memory settings.

4.3.1 vSphere Memory Management Concepts

vSphere virtualizes guest physical memory by adding an extra level of address translation. Hardware-assisted virtualization technologies can make it possible to provide this additional translation with little or no overhead. Managing memory in the hypervisor enables the following:

- Memory sharing across virtual machines that have similar data (same guest operating systems).
- Memory overcommitment, which means allocating more memory to virtual machines than is physically available on the vSphere host. Overcommitment is not necessarily to be avoided. Many customers can achieve high levels of consolidation and efficiency using it. However, overcommitment must be carefully monitored to avoid negative performance impact.
- A memory balloon technique, whereby virtual machines that do not need all memory they have been allocated give memory to virtual machines that require additional allocated memory.

For more details about vSphere memory management concepts, see one of the following documents:

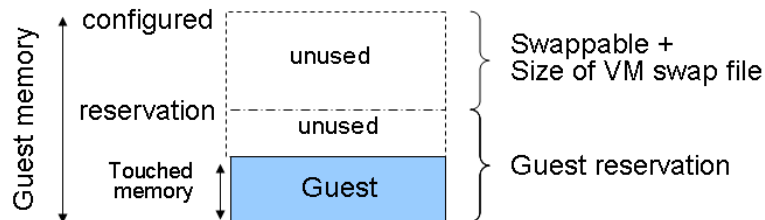
- *vSphere Resource Management Guide* for vSphere 4.x (http://www.vmware.com/pdf/vsphere4/r40_u1/vsp_40_u1_resource_mgmt.pdf).

- *vSphere Resource Management* for vSphere 5.0 (<http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-50-resource-management-guide.pdf>).

4.3.2 Virtual Machine Memory Concepts

The following figure illustrates the use of memory settings parameters in the virtual machine.

Figure 7. Virtual Machine Memory Settings



The vSphere memory settings for a virtual machine include the following parameters:

- Configured memory – memory size of the virtual machine assigned at creation.
- Touched memory – memory actually used by the virtual machine. vSphere allocates guest operating system memory only on demand.
- Swappable – virtual machine memory that can be reclaimed by the balloon driver or by vSphere swapping. Ballooning occurs before vSphere swapping. If this memory is in use by the virtual machine (touched and in use), the balloon driver causes the guest operating system to swap. Also, this value is the size of the per-virtual machine swap file (.vswp) that is created on the VMware vSphere VMFS (Virtual Machine File System) file.
- If the *balloon driver* is unable to reclaim memory quickly enough, or is disabled, or not installed, vSphere forcibly reclaims memory from the virtual machine using the *VMkernel swap file*.

4.3.3 Overhead Memory on Virtual Machines

Virtual machines require a certain amount of available overhead memory to power on. You should be aware of the amount of this overhead.

The following table lists the amount of overhead memory a virtual machine requires to power on. After a virtual machine is running, the amount of overhead memory it uses might differ from the amount listed in the table.

Table 5. Overhead Memory Required for Power-On

VM Memory (MB)	1 vCPU	2 vCPU	4 vCPU	8 vCPU
256	20.29	24.28	32.23	48.16
1024	25.90	29.91	37.86	53.82
4096	48.64	52.72	60.67	76.78
16384	139.62	143.98	151.93	168.60

For additional details, refer to *vSphere Resource Management* (<http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-50-resource-management-guide.pdf>). The table provides a sample of overhead memory values and does not apply to all possible configurations. You can configure a virtual machine to have up to 64 virtual CPUs, up to 1TB of memory.

4.3.4 Allocating Memory to SQL Server Virtual Machines

One of the most critical system resources for SQL Server is memory. Lack of memory resources for the SQL Server database engine results in increased I/O that is orders of magnitude slower than accessing memory.

4.3.4.1. Sphere general best practices

- To avoid performance latency resulting from remote memory accesses, you should size an SQL Server virtual machine's memory so it is less than the amount available per NUMA node. Both vSphere and SQL Server support NUMA. As with SQL Server, vSphere has intelligent, adaptive NUMA scheduling and memory placement policies that can manage all virtual machines transparently, so you do not need to deal with the complexity of manually balancing virtual machines among nodes.
- vSphere supports large pages in the guest operating system. If the operating system or application can benefit from large pages on a native system, that operating system or application can potentially achieve a similar performance improvement in a virtual machine. The use of large pages results in reduced memory management overhead and can therefore increase hypervisor performance. For details, see *Large Page Performance* (http://www.vmware.com/files/pdf/large_pg_performance.pdf).
- Use the Active memory counter from vSphere with caution. Active memory is the amount of memory that's currently being used by the guest operating system and its applications. SQL Server does its own caching and memory management, so the Active memory counter might not accurately reflect the memory consumption of an SQL Server workload. You should always confirm memory usage of an SQL Server virtual machine by checking memory counters within the guest operating system.
- Do not set memory limits for SQL Server virtual machines. Virtual machine memory allocation target is subject to the virtual machine's memory limit and reservation. vSphere offers features to enable dynamic scalability of virtual machine memory. Setting memory limits can cause unexpected swapping.

4.3.4.2. Tier 1 SQL Server workloads

Achieving adequate performance is the primary goal. Consider setting the memory reservation equal to the provisioned memory, to avoid ballooning or swapping. When calculating the amount of memory to provision for the virtual machine, use the following formulas:

`VM Memory = SQL Max Server Memory + ThreadStack + OS Mem + VM Overhead`

`ThreadStack = SQL Max Worker Threads * ThreadStackSize`

`ThreadStackSize` = 1MB on x86
 = 2MB on x64
 = 4MB on IA64

`OS Mem: 1GB for every 4 CPU Cores`

Use SQL Server memory performance metrics and work with your database administrator to determine the SQL Server maximum server memory size and maximum number of worker threads. Refer to the virtual machine overhead table for virtual machine overhead.

4.3.4.3. Lower-Tier SQL Server workloads

Utilizing hardware resources more efficiently and increasing the consolidation ratio are more important to lower-tier SQL Server workloads. Consider the following guideline to maximize resource usages.

- Enable Memory Page Sharing and Memory Ballooning. vSphere provides optimizations such as memory sharing and memory ballooning to reduce the amount of physical memory used on the underlying host. In some cases these optimizations can save more memory than is taken up by the virtualization overhead.
- Confirm that cumulative physical memory available on a server is adequate to meet the needs of the virtual machines by testing target workloads in the virtualized environment. Memory overcommitment should not adversely affect virtual machine performance as long as the actual virtual machine memory requirements are less than the total memory available on the system.

4.4 Storage Configuration Guidelines

Storage configuration is critical to any successful database deployment, especially in virtual environments where you may consolidate many different SQL Server workloads on a single vSphere host. Your storage subsystem should provide sufficient I/O throughput as well as storage capacity to accommodate the cumulative needs of all virtual machines running on your vSphere hosts.

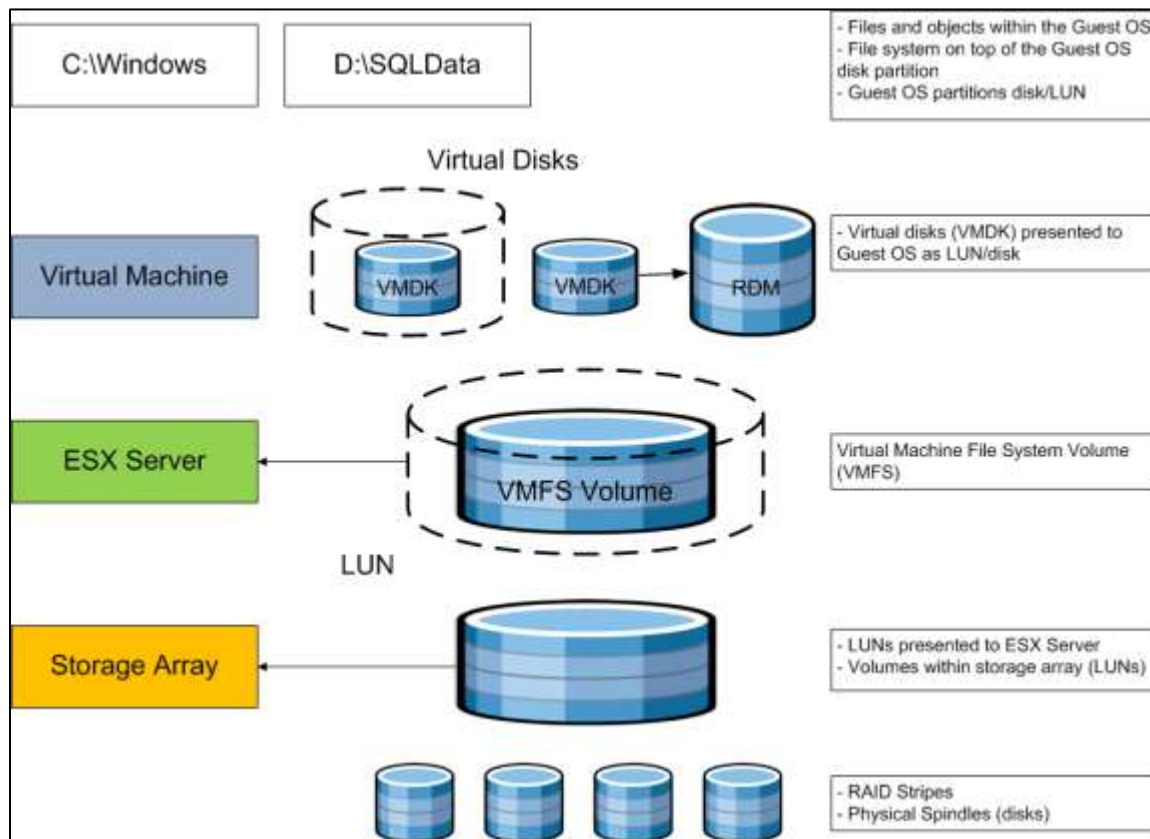
Most traditional physical SQL Server environments created many islands of information. When you move to virtualized SQL Server deployments, a shared storage model strategy provides many benefits, such as more effective storage resource utilization, reduced storage white space, better provisioning, and mobility using vSphere vMotion and VMware vSphere Storage vMotion®.

Most SQL Server performance issues in virtual environments can be traced to improper storage configuration. For information about best practices for SQL Server storage configuration, refer to *Storage Top Ten Practices* (<http://technet.microsoft.com/en-us/library/cc966534.aspx>). Follow these recommendations along with the best practices in this guide.

4.4.1 Storage Virtualization Concepts

As illustrated in the following figure, VMware storage virtualization can be categorized into three layers of storage technology. The storage array is the bottom layer, consisting of physical disks presented as logical disks (storage array volumes or LUNs) to the layer above, with the virtual environment occupied by vSphere. Storage array LUNs that are formatted as VMFS volumes in which virtual disks can be created. Virtual machines consist of virtual disks that are presented to the guest operating system as disks that can be partitioned and used in file systems.

Figure 8. VMware Storage Virtualization Stack



4.4.1.1. VMware Virtual Machine File System (VMFS)

VMFS is a cluster file system that provides storage virtualization optimized for virtual machines. Each virtual machine is encapsulated in a small set of files and VMFS is the default storage system for these files on physical SCSI disks and partitions. VMware supports Fibre Channel and iSCSI protocols for VMFS.

4.4.1.2. Raw Device Mapping

VMware also supports Raw Device Mapping (RDM). RDM allows a virtual machine to directly access a volume on the physical storage subsystem, and can only be used with Fibre Channel or iSCSI. RDM can be thought of as providing a symbolic link from a VMFS volume to a raw volume. The mapping makes volumes appear as files in a VMFS volume. The mapping file, not the raw volume, is referenced in the virtual machine configuration.

4.4.2 VMFS versus RDM for SQL Server

The following sections describe performance and functionality when comparing VMFS and RDM as used with SQL Server.

4.4.2.1. Performance

Customer is often wondering whether VMFS or RDM offers better performance. This depends on the specific workload, and this section and the associated white paper can help in the decision-making process.

From a performance perspective, both VMFS and RDM volumes can provide similar transaction throughput. The following table summarizes some performance testing. For more details, see *Performance Characterization of VMFS and RDM Using a SAN* (http://www.vmware.com/files/pdf/performance_char_vmfs_rdm.pdf).

Figure 9. Random Mixed (50% read/50% write) I/O Operations per Second (higher is better)

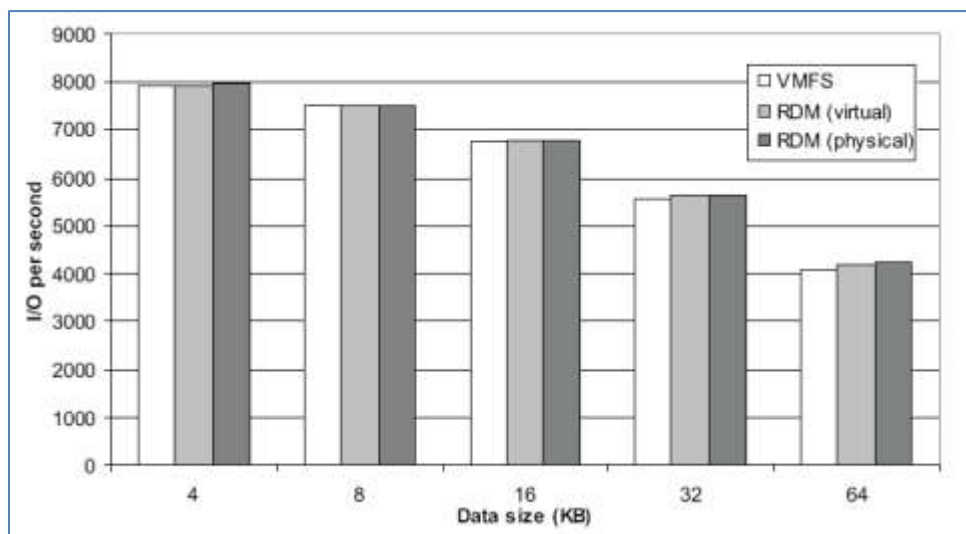
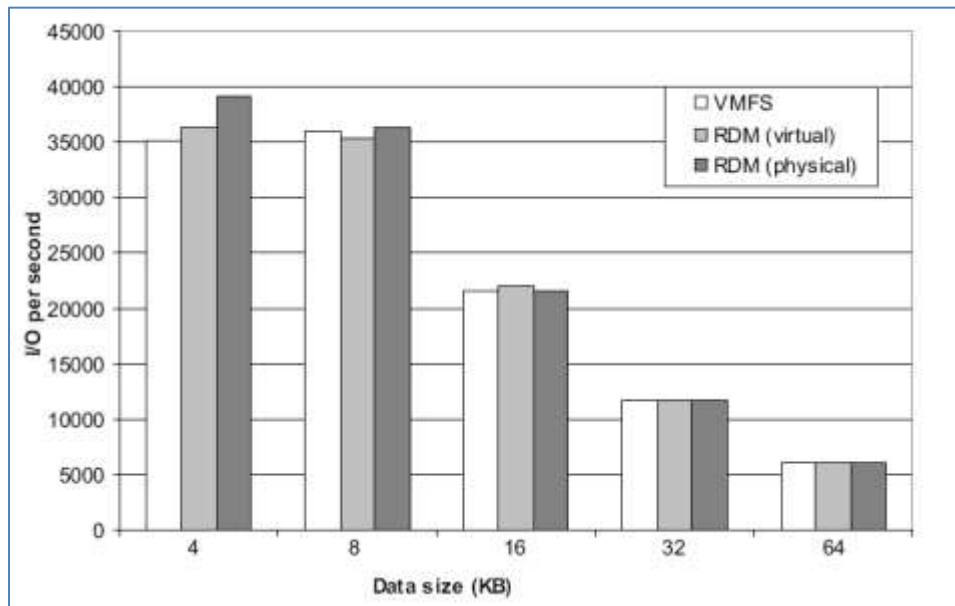


Figure 10. Sequential Read I/O Operations per Second (higher is better)



4.4.2.2. Functionality

There are no set recommendations for using VMFS or RDM in SQL Server deployments, but the following table summarizes some of the options and trade-offs. For a more complete discussion, see the *VMware SAN System Design and Deployment Guide* (http://www.vmware.com/pdf/vsp_4_san_design_deploy.pdf).

Table 6. VMFS and Raw Disk Mapping Trade-Offs

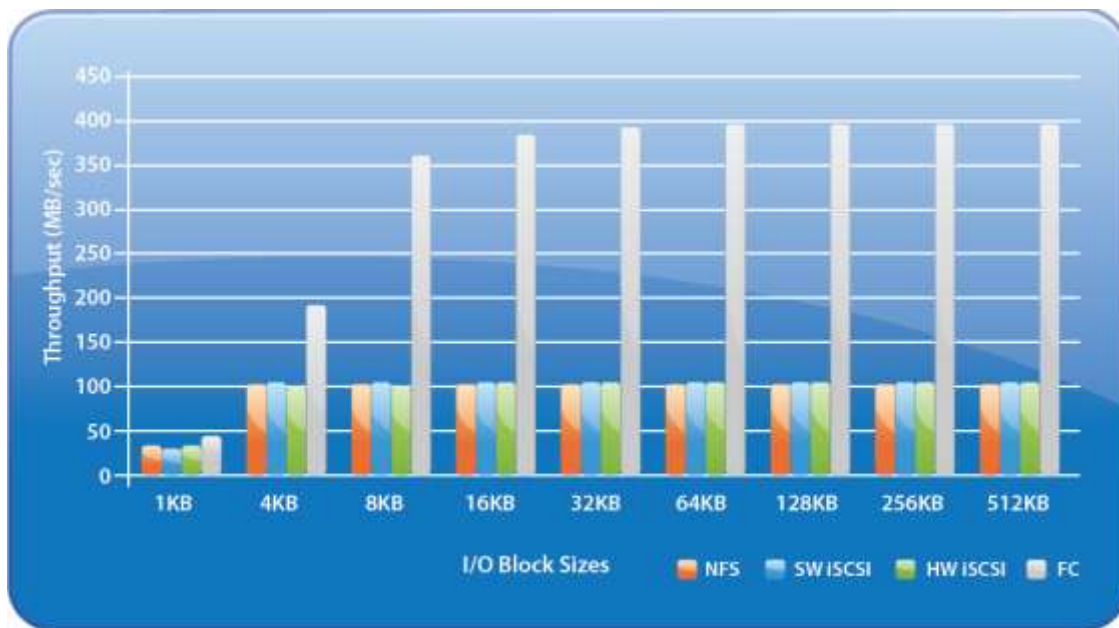
VMFS	RDM
<ul style="list-style-type: none"> Volume can host many virtual machines (or can be dedicated to one virtual machine). Increases storage utilization, provides better flexibility, easier administration and management. Large third-party ecosystem with V2P products to aid in certain support situations. Fully supports VMware vCenter Site Recovery Manager™. Does not support SQL Server failover cluster for shared disk in a multihost cluster setup 	<ul style="list-style-type: none"> Maps a single LUN to one virtual machine so only one virtual machine is possible per LUN. More LUNs are required, so it is more likely to reach the LUN limit of 256 that can be presented to a vSphere host. Uses RDM to leverage array-level backup and replication tools integrated with SQL Server databases. RDM volumes can help facilitate migrating physical SQL Server instances to virtual machines. Required for SQL Server failover cluster for shared disk in a multi-host cluster setup. Fully supports vCenter Site Recovery Manager. Host with large number of RDM LUNs may take longer time to boot or during LUN rescan. For information available at http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1016106.

4.4.3 Storage Protocol Selection

Three storage protocol options are available to SQL Servers running on vSphere: Fibre Channel, iSCSI, and NFS. Wire speed is the limiting factor for I/O throughput when comparing the storage protocols. vSphere can reach link speeds in a single virtual machine environment, and can maintain the throughput for up to 32 concurrent virtual machines for each storage connection option supported. For details, see *Comparison of Storage Protocol Performance in VMware vSphere 4* (http://www.vmware.com/files/pdf/perf_vsphere_storage_protocols.pdf). Fibre Channel might provide maximum I/O throughput, but iSCSI and NFS might offer a better price-performance ratio.

The following figure shows, for each of the storage protocols, the sequential read throughput (in MB/sec) of running a single virtual machine in a standard workload configuration for different I/O block sizes. Refer to *Comparison of Storage Protocol Performance in VMware vSphere 4* (http://www.vmware.com/files/pdf/perf_vsphere_storage_protocols.pdf).

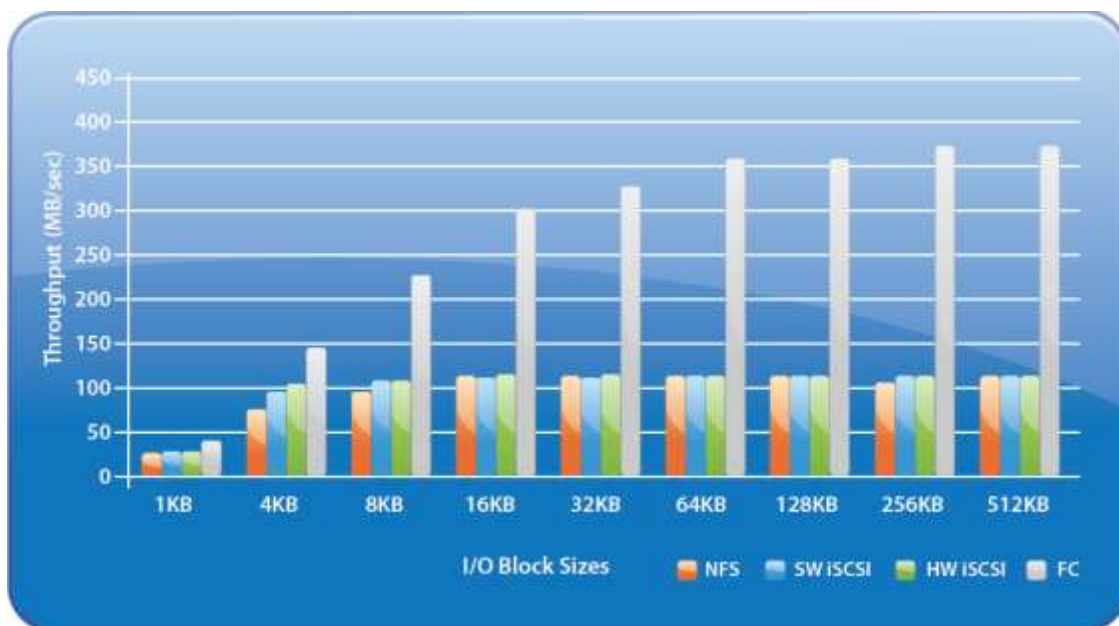
Figure 11. Read Throughput for Different I/O Block Sizes



For Fibre Channel, read throughput is limited by the bandwidth of the 4Gbps Fibre Channel link for I/O sizes at or above 64KB. For IP-based protocols, read throughput is limited by the bandwidth of the 1Gbps Ethernet link for I/O sizes at or above 32KB.

The following figure shows, for each of the storage protocols, the sequential write throughput (in MB/sec) of running a single virtual machine in a standard workload configuration for different I/O block sizes. Refer to *Comparison of Storage Protocol Performance in VMware vSphere 4* (http://www.vmware.com/files/pdf/perf_vsphere_storage_protocols.pdf).

Figure 12. Write Throughput for Different I/O Block Sizes



For Fibre Channel, the maximum write throughput for any I/O block size is consistently lower than read throughput of the same I/O block size. This is the result of disk write bandwidth limitations on the storage array. For the IP-based protocols, write throughput for block sizes at or above 16KB is limited by the bandwidth of the 1Gbps Ethernet link.

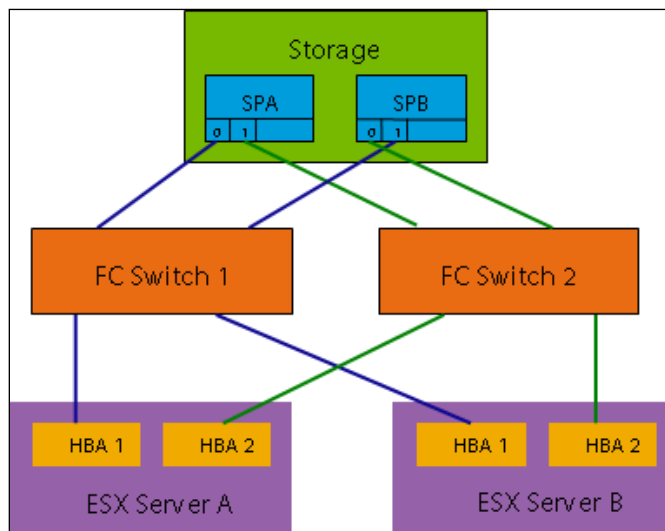
To summarize, a single Iometer thread running in a virtual machine can saturate the bandwidth of the respective networks for all four storage protocols, for both read and write. Fibre Channel throughput performance is higher because of the higher bandwidth of the Fibre Channel link. For the IP-based protocols, there is no significant throughput difference for most block sizes.

4.4.4 Storage Multipathing

It is preferable to deploy virtual machine files on shared storage to take advantage of vSphere vMotion, vSphere HA, and vSphere DRS. This is considered a best practice for mission-critical SQL Server deployments that are often installed on third-party, shared-storage management solutions.

VMware recommends that you set up a minimum of four paths from a vSphere host to a storage array. This means that each host requires at least two HBA ports.

Figure 13. Storage Multipathing Requirements for vSphere



The terms used in the preceding figure are:

- *HBA (Host Bus Adapter)* – A device that connects one or more peripheral units to a computer and manages data storage and I/O processing.
- *FC (Fibre Channel)* – A gigabit-speed networking technology used to build storage area networks (SANs) and to transmit data.
- *SP (Storage Processor)* – A SAN component that processes HBA requests routed through an FC switch and handles the RAID/volume functionality of the disk array.

4.4.5 Allocating Storage for SQL Server Virtual Machines

Most SQL Server performance issues in virtual environments can be traced to improper storage configuration. SQL Server workloads are generally I/O heavy, and a misconfigured storage subsystem can increase I/O latency and significantly degrade performance of SQL Server.

4.4.5.1. Partition Alignment

Aligning file system partitions is a well-known storage best practice for database workloads. Partition alignment on both physical machines and VMware VMFS partitions prevents performance I/O degradation caused by unaligned I/O. An unaligned partition results in additional I/O operations, incurring penalty on latency and throughput, as shown in the following figures. vSphere 5.0 automatically aligns VMFS3 or VMFS5 partitions along a 1MB boundary. If a VMFS3 partition was created using an earlier version of vSphere that aligned along a 64KB boundary, and that file system is then upgraded to VMFS5, it will retain its 64KB alignment. 1MB alignment can be obtained by deleting the partition and recreating it using the vSphere Client and a vSphere 5.0 host.

It is considered a best practice to:

- Create VMFS partitions from within VMware vCenter. They are aligned by default.
- Align the data disk for heavy I/O workloads using `diskpart` or, with Windows Server 2008, the disk is automatically aligned to a 1 MB boundary.
- Consult with the storage vendor for alignment recommendations on their hardware.

For more information about this topic see the white paper *Performance Best Practices for VMware vSphere 5.1* (http://www.vmware.com/pdf/Perf_Best_Practices_vSphere5.1.pdf).

4.4.5.2. Avoid Lazy Zeroing

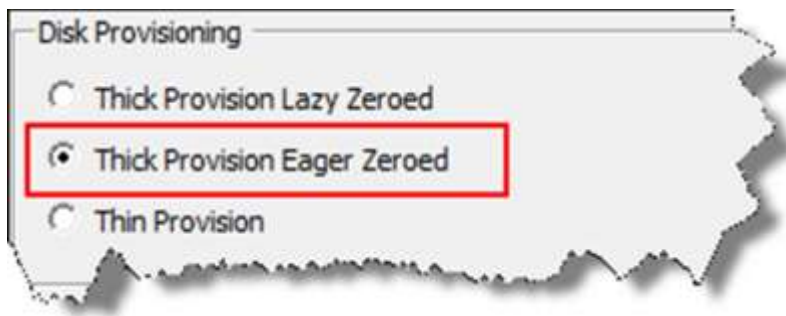
When running on VMFS, virtual machine disk files can be deployed in three different formats: thin, zeroedthick, and eagerzeroedthick. Thin provisioned disk enables 100% storage on demand, where disk space is allocated and zeroed at the time disk is written. This is the vSphere default. Zeroedthick disk storage is pre-allocated, but blocks are zeroed by the hypervisor the first time the disk is written. Eagerzeroedthick disk is pre-allocated and zeroed when the disk is initialized during provision time. There is no additional cost for zeroing the disk at run time.

Both thin and thick options employ a lazy zeroing technique to allow for more efficient disk space usage. However that does not come without a cost, because there is a performance overhead during first write of the disk. Depending on the SQL Server configuration and the type of workloads, the performance could be significant. Write intensive workloads and database maintenance tasks tend to be penalized more by lazy zeroing. If you are deploying a Tier 1 mission-critical SQL Server, VMware recommends that you consider using eagerzeroedthick disks for SQL Server data, transaction log, and tempdb files. An eagerzeroedthick disk can be configured by enabling the **Support clustering features such as Fault Tolerance** option on the **Disk Provisioning** screen in vSphere 4.x or by selecting the **Thick Provision Eager Zeroed** option in vSphere 5.x.

Figure 14. Configure EagerZeroedThick Disk in vSphere 4.x



Figure 15. Configure EagerZeroedThick Disk in vSphere 5.x



4.4.5.3. Optimize with Device Separation

SQL Server files have different disk access patterns as shown in the following table.

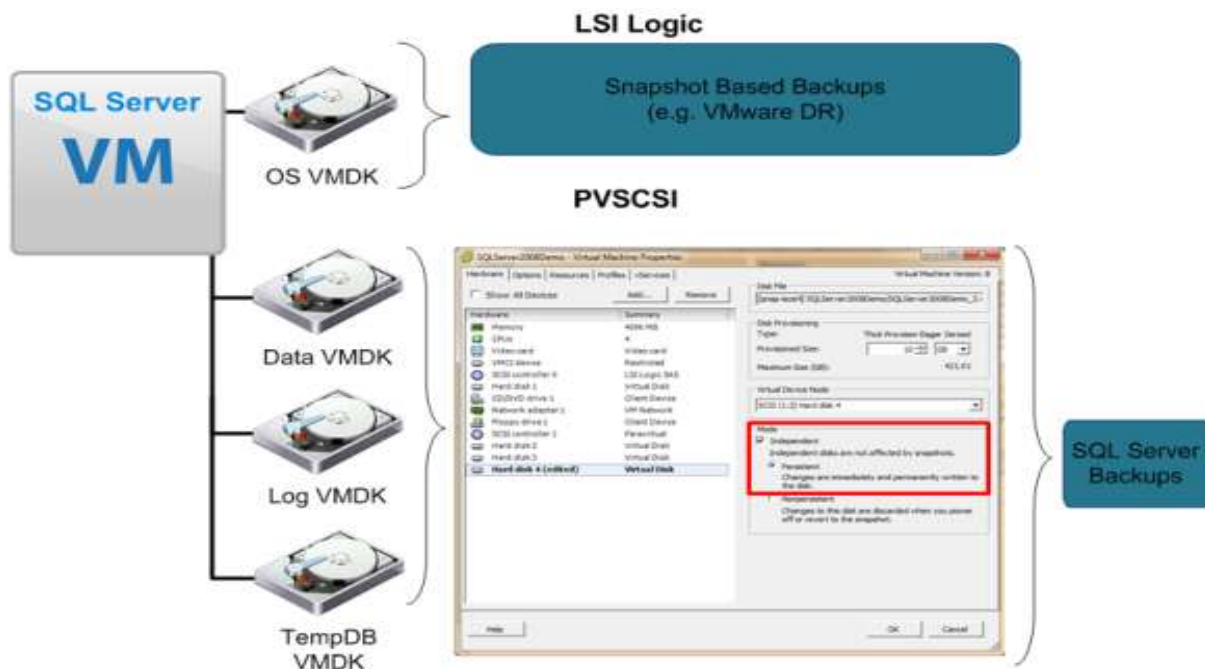
Table 7. Typical SQL Server Disk Access Patterns

Operation	Random / Sequential	Read / Write	Size Range
OLTP – Log	Sequential	Write	Up to 64K
OLTP – Data	Random	Read/Write	8K
Bulk Insert	Sequential	Write	Any multiple of 8K up to 256K
Read Ahead (DSS, Index Scans)	Sequential	Read	Any multiple of 8KB up to 512K
Backup	Sequential	Read	1 MB

When deploying a Tier 1 mission-critical SQL Server, placing SQL Server binary, data, transaction log, and tempdb files on separate storage devices allows for maximum flexibility, and improves performance. The following guidelines can help to achieve best performance:

- Place SQL Server binary, log, and data files into separate VMDKs. In addition to the performance advantage, separating SQL Server binary from data and log also provides better flexibility for backup. The OS/SQL Server Binary VMDK can be backed up with snapshot-based backups, such as VMware Data Recovery. The SQL Server data and log files can be backed up through traditional database backup solutions.
- Maintain 1:1 mapping between VMDKs and LUNs. When this is not possible, group VMDKs and SQL Server files with similar I/O characteristics on common LUNs.
- Use multiple vSCSI adapters. Place SQL Server binary, data, log onto separate vSCSI adapter optimizes I/O by distributing load across multiple target devices.

Figure 16. Optimize with Device Separation



For lower-tier SQL Server workloads, consider the following:

- Deploying multiple, lower-tier SQL Server systems on VMFS facilitates easier management and administration of template cloning, snapshots, and storage consolidation.
- Manage performance of VMFS. The aggregate IOPS demands of all virtual machines on the VMFS should not exceed the IOPS capability the physical disks.
- Use VMware vSphere Storage DRS™ for automatic load balancing between datastores to provide space and avoid I/O bottlenecks as per pre-defined rules.

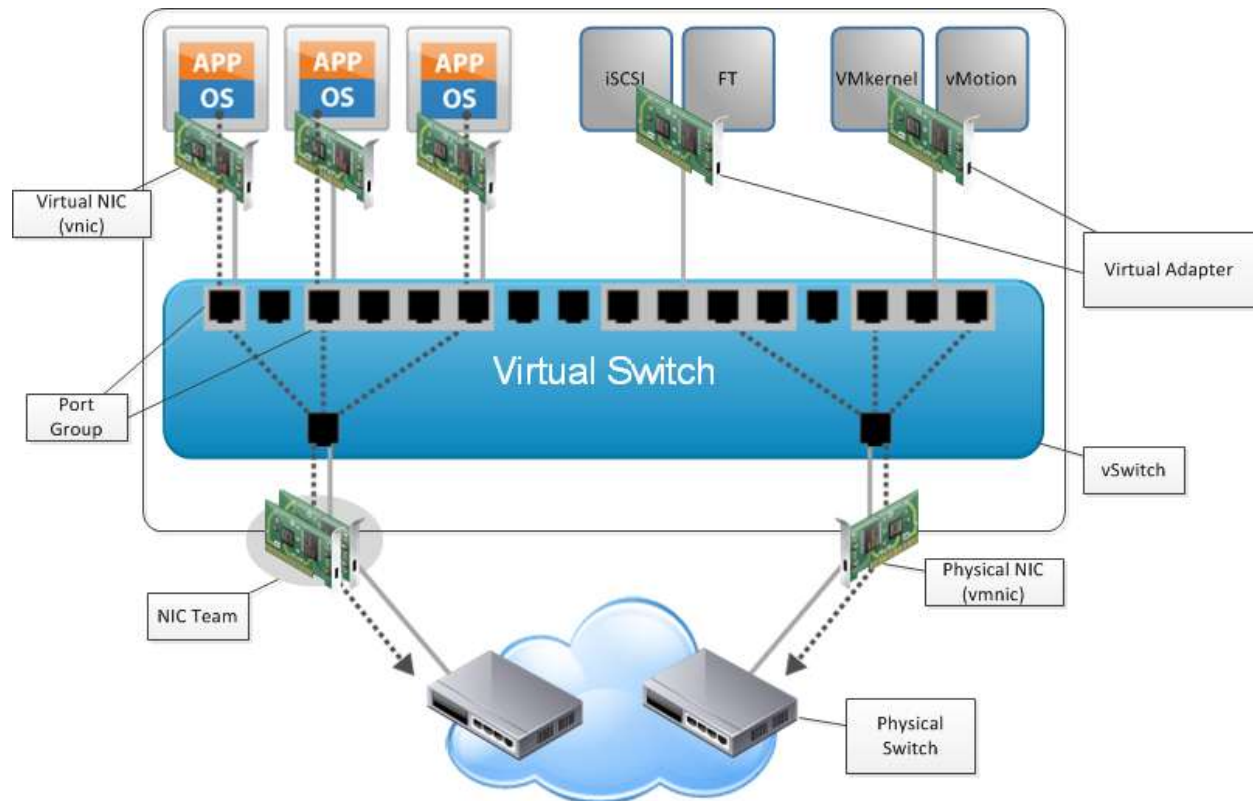
4.5 Networking Configuration Guidelines

Networking in the virtual world follows the same concepts as in the physical world, but these concepts are applied in software instead of using physical cables and switches. Many of the best practices that apply in the physical world continue to apply in the virtual world, but there are additional considerations for traffic segmentation, availability, and making sure that the throughput required by services hosted on a single server can be fairly distributed.

4.5.1 Virtual Networking Concepts

The following figure provides a visual overview of the components that make up the virtual network.

Figure 17. Virtual Networking Concepts



As shown in the figure, the following components make up the virtual network:

- Physical switch – vSphere host-facing edge of the local area network.
- Physical network interface (vmnic) – Provides connectivity between the vSphere host and the local area network.
- vSwitch – The virtual switch is created in software and provides connectivity between virtual machines. Virtual switches must uplink to a physical NIC (also known as vmnic) to provide virtual machines with connectivity to the LAN. Otherwise, virtual machine traffic is contained within the virtual switch.
- Port group – Used to create a logical boundary within a virtual switch. This boundary can provide VLAN segmentation when 802.1q trunking is passed from the physical switch, or can create a boundary for policy settings.
- Virtual NIC (vNIC) – Provides connectivity between the virtual machine and the virtual switch.

- Virtual Adapter – Provides Management, vSphere vMotion, and FT Logging when connected to a vSphere Distributed Switch.
- NIC Team – Group of physical NICs connected to the same physical/logical networks providing redundancy.

4.5.2 Virtual Networking Best Practices

Depending on the specific on the SQL Server workload, some workloads are more sensitive to network latency than others. To configure the network for your SQL Server virtual machine, start with a thorough understanding of your workload network requirements. Monitoring the following performance metrics on the existing workload for a representative period using Windows Perfmon or VMware Capacity Planner™ can easily help determine the requirements for an SQL Server virtual machine.

The following guidelines generally apply to provisioning the network for an SQL Server virtual machine.

- Use NIC Teaming – Use two physical NICs per vSwitch, and if possible, uplink the physical NICs to separate physical switches. Teaming provides redundancy against NIC failure and, if connected to separate physical switches, against switch failures. NIC teaming does not necessarily provide higher throughput.
- If a physical NIC is shared by multiple consumers (that is, virtual machines and/or the VMkernel), each such consumer could impact the performance of others. Thus, for the best network performance, use separate physical NICs for management traffic (vSphere vMotion, FT logging, VMkernel) and virtual machine traffic. 10Gbps NICs have multiqueue support. 10Gbps NICs can separate these consumers onto different queues. Thus as long as 10Gbps NICs have sufficient queues available, this recommendation typically does not apply.
- If using iSCSI, the network adapters should be dedicated to either network communication or iSCSI, but not both.
- Enable jumbo frames for iSCSI and the vSphere vMotion network.
- Use the VMXNET3 paravirtualized NIC. VMXNET 3 is the latest generation of paravirtualized NICs designed for performance. It offers several advanced features including multiqueue support, Receive Side Scaling, IPv4/IPv6 offloads, and MSI/MSI-X interrupt delivery.

5. SQL Server In-Guest Best Practices

5.1 Maximum Server Memory and Minimum Server Memory

SQL Server can dynamically adjust memory consumption based on workloads. SQL Server maximum server memory and minimum server memory configuration settings allow you to define the range of memory for the SQL Server process in use. The default setting for minimum server memory is 0, and the default setting for maximum server memory is 2147483647MB. Minimum server memory will not immediately be allocated on startup. However, after memory usage has reached this value due to client load, SQL Server will not free memory unless the minimum server memory value is reduced.

SQL Server is capable of consuming all memory on the virtual machine. Setting the maximum server memory allows you to reserve sufficient memory for the operating system and other applications running on the virtual machine. In a traditional SQL Server consolidation scenario where you are running multiple instances of SQL Server on the same virtual machine, setting maximum server memory will allow memory to be shared effectively between the instances.

Setting the minimum server memory is a good practice to maintain SQL Server performance under host memory pressure. When running SQL Server on vSphere, if the vSphere host is under memory pressure, the balloon drive might inflate and take memory back from the SQL Server virtual machine. Setting the minimum server memory provides SQL Server with at least a reasonable amount of memory.

For Tier 1 mission-critical SQL Server deployments, consider setting the SQL Server memory to a fixed amount by setting both maximum and minimum server memory to the same value. Before setting the maximum and minimum server memory, confirm that adequate memory is left for the operating system and virtual machine overhead.

For performing SQL Server maximum server memory sizing for vSphere, use the following formulas as a guide:

$$\text{SQL Max Server Memory} = \text{VM Memory} - \text{ThreadStack} - \text{OS Mem} - \text{VM Overhead}$$
$$\text{ThreadStack} = \text{SQL Max Worker Threads} * \text{ThreadStackSize}$$
$$\text{ThreadStackSize} = 1\text{MB on x86}$$
$$= 2\text{MB on x64}$$
$$= 4\text{MB on IA64}$$

OS Mem: 1GB for every 4 CPU Cores

5.2 Lock Pages in Memory

Granting the Lock Pages in Memory user right to the SQL Server service account prevents SQL Server buffer pool pages from paging out by Windows. This setting is useful and has a positive performance impact because it prevents Windows from paging a significant amount of buffer pool memory out of the process, which enables SQL Server to manage the reduction of its own working set.

Any time Lock Pages in Memory is used, because SQL Server memory is locked and cannot be paged out by Windows, you might experience negative impacts if the vSphere balloon driver is trying to reclaim memory from the virtual machine. If you set the SQL Server Lock Pages in Memory user right, also set the virtual machine's reservations to match the amount of memory you set in the virtual machine configuration.

Consider setting the Lock Pages in Memory user right and setting virtual machine memory reservations to improve the performance and stability of your SQL Server running vSphere if you are deploying a Tier 1 mission-critical SQL Server installation. Setting virtual machine memory reservations prevent the balloon driver from inflating into the SQL Server virtual machine's memory space. For instructions on enabling Lock Pages in Memory, refer to *Enable the Lock Pages in Memory Option (Windows)* (<http://msdn.microsoft.com/en-us/library/ms190730.aspx>). Lock Pages in Memory should also be used in conjunction with the Max Server Memory setting to avoid SQL Server taking over all memory on the virtual machine.

For lower-tiered SQL Server workloads where performance is less critical, the ability to overcommit to maximize usage of the available host memory might be more important. When deploying lower-tiered SQL Server workloads, VMware recommends that you do not enable the Lock Pages in Memory user right. Lock Pages in Memory causes conflicts with vSphere balloon driver. For lower tier SQL workloads, it is better to have balloon driver manage the memory dynamically. Having balloon driver dynamically manage vSphere memory can help maximize memory usage and increase consolidation ratio.

5.3 Large Pages

Hardware assist for MMU virtualization typically improves the performance for many workloads. However, it can introduce overhead arising from increased latency in the processing of TLB misses. This cost can be eliminated or mitigated with the use of large pages. Refer to *Large Page Performance* <http://www.vmware.com/resources/techresources/1039> for additional information.

SQL Server supports the concept of large pages when allocating memory for some internal structures and the buffer pool, when the following conditions are met:

- You are using SQL Server Enterprise Edition.
- The computer has 8GB or more of physical RAM.
- The Lock Pages in Memory privilege is set for the service account.

As of SQL Server 2008, some of the internal structures, such as lock management and buffer hash, can use large pages automatically if the preceding conditions are met. You can confirm that by checking the ERRORLOG for the following messages:

```
2009-06-04 12:21:08.16 Server Large Page Extensions enabled.
2009-06-04 12:21:08.16 Server Large Page Granularity: 2097152
2009-06-04 12:21:08.21 Server Large Page Allocated: 32MB
```

On 64-bit system, you can further enable all SQL Server buffer pool memory to use large pages by starting SQL Server with trace flag 834. Consider the following behavior changes when you enable trace flag 834:

- With large pages enabled in the guest operating system, and the virtual machine is running on a host that supports large pages, vSphere does not perform Transparent Page Sharing on the virtual machine's memory.
- With trace flag 834 enabled, SQL Server startup behavior changes. Instead of allocating memory dynamically at runtime, SQL Server allocates all buffer pool memory during startup. Therefore, SQL Server startup time can be significantly delayed.
- With trace flag 834 enabled, SQL Server allocates memory in 2MB contiguous blocks instead of 4KB blocks. After the host has been running for a long time, it might be difficult to obtain contiguous memory due to fragmentation. If SQL Server is unable to allocate the amount of contiguous memory it needs, it can try to allocate less, and SQL Server might then run with less memory than you intended.

Although trace flag 834 improves the performance of SQL Server, it might not be suitable for use in all deployment scenarios. With SQL Server running in a highly consolidated environment, this setting is not recommended. This setting is more suitable for a high performance Tier 1 SQL Server workloads where there is no oversubscription of the host, and no overcommitment of memory. Always confirm that the correct large pages memory is granted by checking messages in the SQL Server `ERRORLOG`. See the following example:

```
2009-06-04 14:20:40.03 Server Using large pages for buffer pool.  
2009-06-04 14:27:56.98 Server 8192 MB of large page memory allocated.
```

Refer to *SQL Server and Large Pages Explained* (<http://blogs.msdn.com/b/psssql/archive/2009/06/05/sql-server-and-large-pages-explained.aspx>) for additional information on running SQL Server with large pages.

5.4 Using Virus Scanners on SQL Server

Customers might have virtual scan software running on an SQL Server virtual machine. However, you can consider excluding SQL Server data and log files from real time virus monitoring and scanning to minimize performance impacts to SQL Server. The database server should be secured by other means so that it is not vulnerable to malware. This includes both strict access controls and operational discipline such as not using an Internet browser to download data or executable files from external sites.

6. Ongoing Performance Monitoring and Tuning

Virtualization adds new software layers and new types of interactions between the database and hardware components. Although the general methodology for monitoring and troubleshooting database performance does not change, VMware provides additional tools for monitoring and troubleshooting at the physical host level.

6.1 Performance Monitoring Tools

When monitoring database performance, you can use the database virtual machine-level performance monitoring tools as the primary tools for identifying problem areas and resource bottlenecks. The methodologies are the same as performance monitoring for a physical database server. VMware provides additional tools for host-level performance monitoring. You can further correlate performance data collected from the database virtual machine level with data from the host level. By focusing on key performance metrics you can quickly isolate issues to a particular resource area.

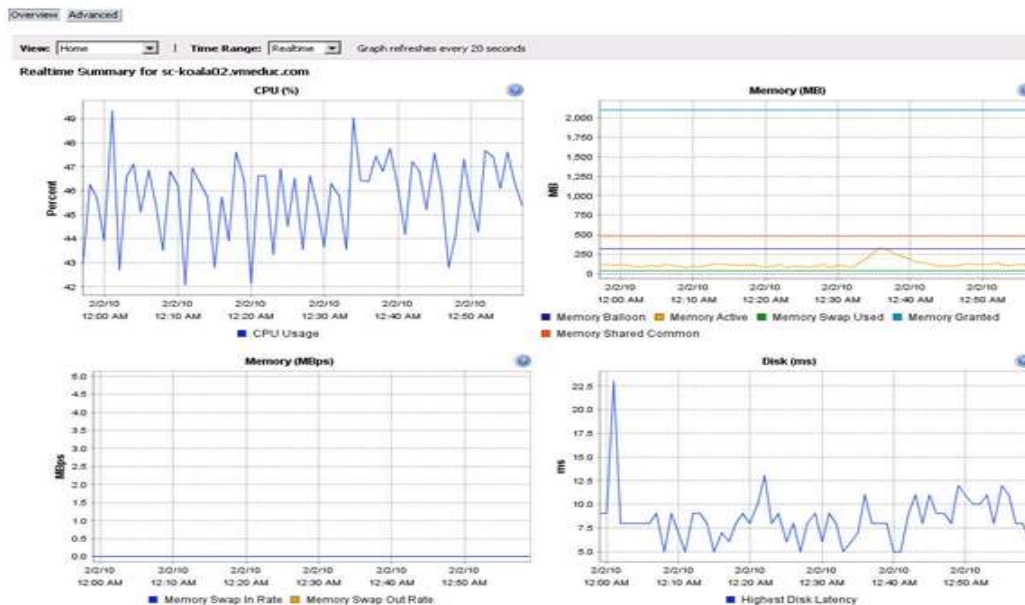
6.1.1 Host-Level Monitoring Tools

vSphere provides two main tools for observing and collecting performance data: the VMware vSphere Client™, and the `resxtop` utilities.

6.1.1.1 vSphere Client

The vSphere Client is a graphical interface tool that connects to a vSphere host to display real-time performance data about the host and virtual machines. When connected to a vCenter host, the vSphere Client can also display historical performance data about all hosts and virtual machines managed by that server. The vSphere Client is easy to use, provides access to the most important configuration and performance information, and does not require high levels of privilege to access performance data. The vSphere Client is the primary tool for observing daily performance and configuration data for vSphere hosts.

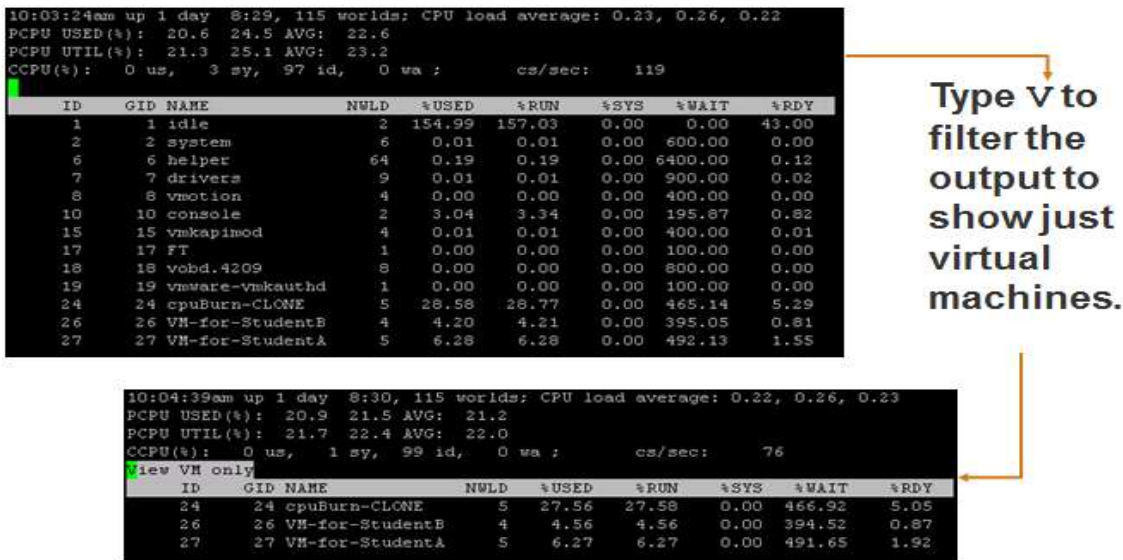
Figure 18. Performance Chart Viewed with the vSphere Client



6.1.1.2. resxtop

The `resxtop` utility provides access to detailed performance data from a vSphere host. Beyond the performance metrics available through the vSphere Client, these utilities provide access to advanced performance metrics that are not available elsewhere. Using `resxtop`, you can view a large amount of available data and quickly observe a large number of performance metrics. However, the `resxtop` tool requires root-level access privileges to the vSphere host. Use the `resxtop` utility for advanced troubleshooting when more detailed performance data is needed.

Figure 19. `resxtop` CPU Metrics



6.1.2 SQL Server Monitoring Tools

SQL Server provides tools and utilities to assist with performance monitoring and tuning. These include SQL Server Perfmon counters, SQL Server Profiler, and SQL Server Dynamic Management Views (DMVs). In a virtualized database environment, these are still the primary tools for monitoring the internal health of the SQL Server system. However, DBAs should be aware that any time-based measurements reported from a virtual machine-level monitoring tool might not be accurate. The accuracy of in-guest tools depends on the guest operating system and kernel version being used, and the total load of the vSphere host.

For effective monitoring and troubleshooting of a database performance problem in VMware, focus on identifying performance bottlenecks instead of time-base measurements. For example, to troubleshoot an SQL Server performance issue, you might focus on the Process Queue Length for identifying a CPU bottleneck instead of looking at the counter for % Processor Time. Similarly, you can effectively identify a disk bottleneck if the Disk Queue Length is high and a large number of SQL Server users are waiting on `PAGEIOLATCH_EX`, `PAGEIOLATCH_SH`. After a resource bottleneck is identified on the database virtual machine, you can then correlate statistics collected from the vSphere Client and `resxtop` to identify any host level configuration issue that affects database virtual machine performance.

The following table shows the key SQL Server counters to monitor.

Table 8. Key SQL Counters

Resource	Metric	Description
SQLServer :SQL Statistics	Batch Requests/sec	SQL Server batch requests per second.
	SQL Compilations/sec	Number of SQL compilations. Compilations/sec Includes both initial compiles and subsequent recompiles.
	SQL Re-Compilations/sec	Number of SQL re-compiles.
SQLServer: Buffer Manager	Buffer cache hit ratio	Percentage of time that the pages requested are already in cache. Available in Capacity Planner.
	Page life expectancy	Time in seconds the data pages, on average, stay in SQL cache. Low page life <300 can indicate (1) SQL cache is cold, (2) memory problems or (3) missing indexes. Correlate to Lazy writes/sec and Checkpoint pages/sec.
	Cache Hit Ratio	Percentage of time that the procedure plan pages are already in cache. For example, procedure cache hits. That is, how frequently a compiled procedure is found in the procedure cache (therefore avoiding the need to recompile).
	Free Pages	Number of pages on all Free Lists. Available in Capacity Planner.
	Procedure Cache Pages	Number of pages used to store compiled queries. Available in Capacity Planner.
	Readahead Pages/sec	When the SQL Server detects sequential access of data or index pages, it reads pages ahead in anticipation of a read request. Available in Capacity Planner.
SQLServer: Memory Manager	Memory Grants Pending	Memory resources are required for each user request. If sufficient memory is not available, the user waits until there is adequate memory for the query to run.
	Connection Memory (KB)	Total amount of dynamic memory the server is using for maintaining connections Available in Capacity Planner.
	Granted Workspace Memory (KB)	Total amount of memory granted to executing processes, which is used for hash sort and create index operations Available in Capacity Planner.
	Lock Memory (KB)	Total amount of dynamic memory the server is using for locks. Available in Capacity Planner.
	SQL Cache memory (KB)	Total amount of dynamic memory the server is using for the dynamic SQL cache Available in Capacity Planner.

Resource	Metric	Description
	Target Server Memory (KB)	Total amount of dynamic memory the server is willing to consume. Available in Capacity Planner.
	Total Server Memory (KB)	Total amount of dynamic memory the server is currently consuming. Available in Capacity Planner.
SQLServer: General Statistics	Logins/sec	Number of logins per second.
	Logout/sec	Number of logouts per second.
	User connections	Number of user connections.
SQLServer: Latches	Average Latch Wait Time (ms)	Latches are short term light weight synchronization object. Latches are not held for the duration of a transaction. Typical latching operations during row transfers to memory, controlling modifications to row offset table, and so on.
SQLServer:Locks	Average Wait Time (ms)	Transactions should be as short as possible to limit the blocking of other users. Available in Capacity Planner.
	Number of deadlocks/sec	Number of lock requests that resulted in a deadlock. Available in Capacity Planner.
SQLServer: Databases	Log Flush Wait Time	Waiting for transaction log writes (ms).
	Log Flush Waits/sec	This is the number of commits waiting on a log flush.
	Transactions /sec	SQL Server transactions per second Available in Capacity Planner.
	Data File(s) Size (KB)	Cumulative size of all data files in the database. Available in Capacity Planner.
	Log File(s) Size (KB)	Cumulative size of all data files in the database. Available in Capacity Planner.

6.2 Key Performance Metrics on vSphere

Among the performance data exposed by vSphere, the following is a list of key metrics that can help DBAs quickly isolate issues to a specific resource area such as CPU, memory, storage, or network. See VMware Communities: *Interpreting resxtop Statistics* (<http://communities.vmware.com/docs/DOC-9279>) and *vCenter Performance Counters* (<http://communities.vmware.com/docs/DOC-5600>) for a full list of counters.

Note The measurement units reported in `resxtop` and the vSphere Client can differ. See the preceding VMware community links for details.

Table 9. Key Performance Metrics

Resource	Metric (resxtop)	Metric (vSphere Client)	Host/Virtual Machine	Description
CPU	%USED	Used	Both	CPU used over the collection interval (%).
	%RDY	Ready	Virtual Machine	CPU time spent in ready state.
	%SYS	System	Both	Percentage of time spent in the vSphere Server VMKernel.
Memory	Swapin, Swapout	Swapinrate, Swapoutrate	Both	Memory vSphere host swaps in/out from/to disk (per virtual machine, or cumulative over host).
	MCTLSZ (MB)	vmmemctl	Both	Amount of memory reclaimed from resource pool by way of ballooning.
Disk	READs/s, WRITEs/s	NumberRead, NumberWrite	Both	Reads and Writes issued in the collection interval.
	DAVG/cmd	deviceLatency	Both	Average latency (ms) of the device (LUN).
	KAVG/cmd	KernelLatency	Both	Average latency (ms) in the VMkernel, also known as queuing time.
	GAVG/cmd	TotalLatency	Both	Average latency (ms) in the guest. GAVG = DAVG + KAVG.
Network	MbRX/s, MbTX/s	Received, Transmitted	Both	Amount of data transmitted per second.
	PKTRX/s, PKTTX/s	PacketsRx, PacketsTx	Both	Packets transmitted per second.
	%DRPRX, %DRPTX	DroppedRx, DroppedTx	Both	Dropped packets per second.

Of the CPU counters, the total used time indicates system load. Ready time indicates overloaded CPU resources. A significant swap rate in the memory counters is a clear indication of a shortage of memory, and high device latencies in the storage section point to an overloaded or misconfigured array. Network traffic is not frequently the cause of most database performance problems.

7. vSphere Enhancements for Deployment and Operations

You can leverage vSphere to provide significant benefits in a virtualized SQL datacenter, including:

- Increased operational flexibility and efficiency – Rapid software applications and services deployment in shorter time frames.
- Efficient change management – Increased productivity when testing the latest Windows and SQL Server software patches and upgrades.
- Minimized risk and enhanced IT service levels – Zero-downtime maintenance capabilities, rapid recovery times for high availability, and streamlined disaster recovery across the datacenter.

7.1 vSphere vMotion, DRS, and vSphere HA

vSphere vMotion technology enables the migration of virtual machines from one physical server to another without service interruption. This migration allows you to move SQL Server virtual machines from a heavily-loaded server to one that is lightly loaded, or to offload them to allow for hardware maintenance without any downtime.

DRS takes the vSphere vMotion capability a step further by adding an intelligent scheduler. DRS allows you to set resource assignment policies that reflect business needs. DRS performs the calculations and automatically handles the details of physical resource assignments. It dynamically monitors the workload of the running virtual machines and the resource utilization of the physical servers within a cluster.

vSphere vMotion and DRS perform best under the following conditions:

- The source and target vSphere hosts must be connected to the same gigabit network and the same shared storage.
- A dedicated gigabit network for vSphere vMotion is recommended.
- The destination host must have enough resources.
- The virtual machine must not use physical devices such as CD-ROM or floppy.
- The source and destination hosts must have compatible CPU models, or migration with vSphere vMotion will fail. For a listing of servers with compatible CPUs, consult vSphere vMotion compatibility guides from specific hardware vendors.
- To minimize network traffic it is best to keep virtual machines that communicate with each other together on the same host machine.
- Virtual machines with smaller memory sizes are better candidates for migration than larger ones.

Note VMware does not currently support vSphere vMotion or DRS load balancing with SQL Server failover cluster. However, a cold migration is possible after the guest operating system is properly shut down.

With vSphere HA, SQL Server virtual machines on a failed vSphere host can be restarted on another vSphere host. This feature provides a cost-effective failover alternative to expensive third-party clustering and replication solutions. If you use vSphere HA, be aware that:

- vSphere HA handles vSphere host hardware failure but does not monitor the status of the SQL Server services—these must be monitored separately.
- Proper DNS hostname resolution is required for each vSphere host in a vSphere HA cluster.
- vSphere HA heartbeat is sent over the vSphere service console network, so redundancy in this network is recommended.

7.2 Templates

VMware template cloning can increase system administration and testing productivity in SQL Server environments. A VMware template is a *golden image* of a virtual machine that can be used as a master copy to create and provision new virtual machines. You can use virtual machine templates create an image of the operating system with service patches, significantly reduced the time needed to provision an SQL Server.

7.3 vFabric Data Director

VMware vFabric™ Data Director 2.5 extends support for SQL Server 2008 R2 and SQL Server 2012. vFabric Data Director helps provision and administer databases more efficiently, more securely, and more cost-effectively.

7.3.1 Standardization

One of the key objectives of vFabric Data Director is to bring order into the database management space. By streamlining many of the repetitive tasks with easy to create templates, vFabric Data Director is able to enforce predefined policies across the enterprise. This in turn allows IT to leverage the underlying High Availability of the underlying virtualized infrastructure to scale dynamically and deliver higher SLAs.

By supporting common security and administration policies across databases, your new DBaaS can leverage common policies across a farm of databases running a variety of operating systems. These policies can enforce uniformity of frequent administration tasks across databases, providing compliance, consistency, and security, and at a lower total cost.

7.3.2 Automation

vFabric Data Director automates the administration and monitoring tasks like creation, backup, recovery, tuning, optimization, patching, and upgrading. Based on custom policies, vFabric Data Director enables DBAs to greatly automate mundane database management tasks and focus on proactive maintenance and dealing with abnormalities.

7.3.3 Self-Service

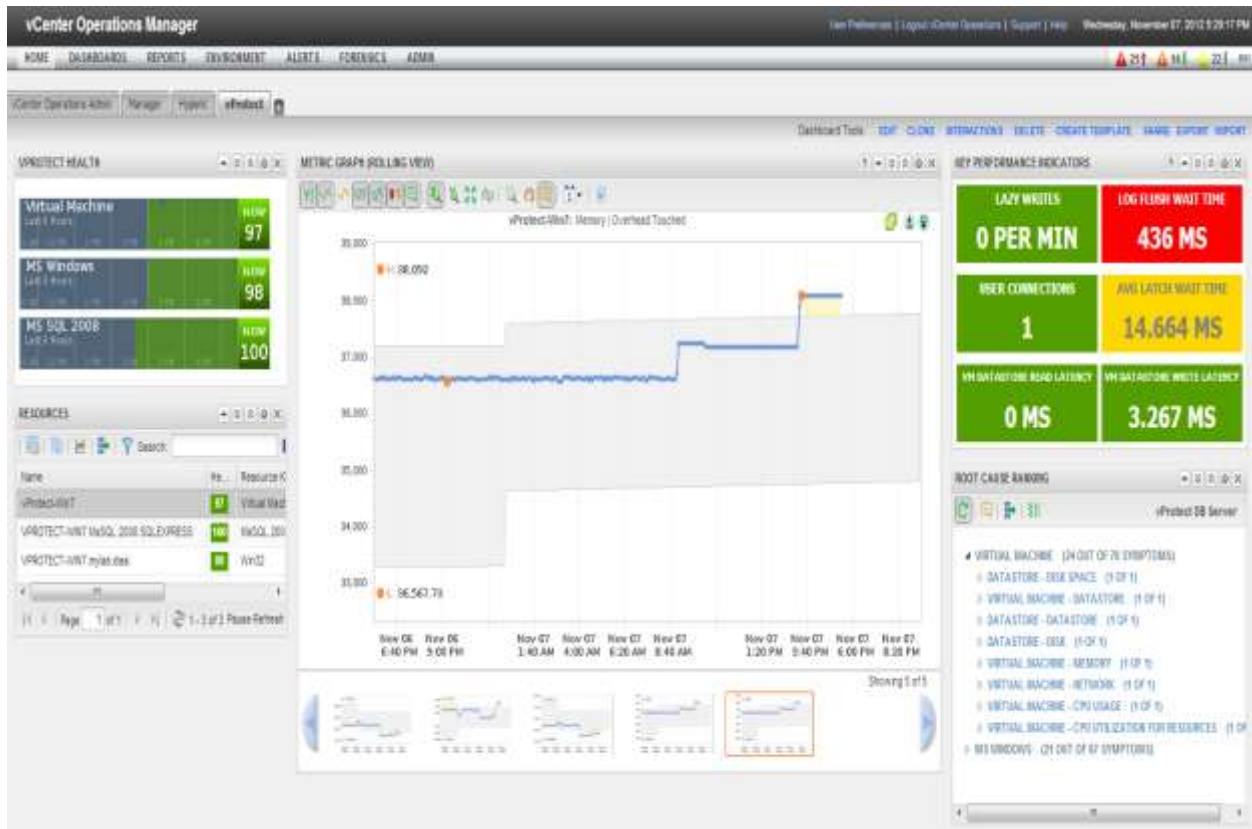
The basic requirement of anything as a service is the capability for users to self-provision necessary resources. vFabric Data Director allows database consumers such as application developers, testers, and architects to provision databases easily based on built-in or custom templates and under fine grained security constraints.

Figure 20. vFabric Data Director

7.1 vCenter Operations Management Suite

The VMware vCenter Operations Management Suite™ can provide a holistic approach to performance, capacity, and configuration management. By using patented analytics, service levels can be monitored and maintained proactively. vCenter Operations Management Suite supports a number of adapters, for example, VMware vFabric Hyperic®, Microsoft System Center Operations Manager, and General SQLLoader Adapter for monitoring SQL Server performance. When performance or capacity problems arise in your SQL Server environment, VMware vCenter Operations Manager™ is able to analyze metrics from the application all the way through to the infrastructure to provide insight into problematic components, whether it is compute (physical or virtual), storage, networking, operating system, or application related. By establishing trends over time, vCenter Operations Manager can cut through the noise of false alerts and proactively alert on the potential root cause of building performance problems before end users are impacted. The following is an example showing vCenter Operations Management Suite using a vFabric Hyperic adapter to display statistics from an SQL Server database.

Figure 21. vCenter Operations Manager



In an SQL Server environment, constant monitoring is required to maintain acceptable service levels. vCenter Operations Management Suite includes patented capacity analytics which can run through “what if” capacity scenarios to understand growth trends and identify upcoming compute power shortages or over-provisioned resources. vCenter Operations Manager monitors configurations across virtual machines and detects unwanted changes to help maintain continuous compliance with operational best practices.

7.2 vCenter Site Recovery Manager

VMware vCenter Site Recovery Manager™ takes advantage of virtual machine encapsulation to make testing and initiating DR failover a simple, integrated vCenter process. vCenter Site Recovery Manager runs alongside VMware vCenter Server™ to provide planning, testing, and automated recovery in the case of a disaster. By using VMware vSphere Replication or storage-based replication technology, vCenter Site Recovery Manager eliminates the manual steps required during a failover scenario to provide consistent and predictable results. At a high level, the steps that can be performed during a failover test or an actual run are as follows:

- Shut down production virtual machines (failover).
- Promote recovery storage to primary (failover).
- Take and mount snapshot of recovery storage in read/write mode (test only).
- Rescan vSphere hosts to make storage visible.
- Register recovery virtual machines.
- Power on virtual machines at recovery site.
- Reconfigure IP settings and update DNS if required.
- Verify that VMware Tools™ starts successfully on recovered virtual machines.
- Power off recovered virtual machines (test only).
- Un-register virtual machines (test only).
- Remove storage snapshot from recovery side (test only).

SQL Server has a variety of options for disaster recovery, including AlwaysOn Availability Groups, database mirroring, failover clustering, and log shipping. While all of these are good choices for SQL Server recovery, the application-centric nature of these technologies might not be in line with a company's disaster recovery plans. vCenter Site Recovery Manager is not a replacement for application-aware clustering solutions that might be deployed within the guest operating system. vCenter Site Recovery Manager provides integration of the storage replication solution, vSphere, and customer-developed scripts to provide a simple, repeatable, and reportable process for disaster recovery of the entire virtual environment, regardless of the application.

Figure 22. VMware vCenter Site Recovery Manager

