Oracle Database 10g Performance on
Windows Server 2008 Hyper-V Virtual Machine

By: Gary W Parker
Performance Tuning Corporation
Introduction

Microsoft’s current server operating system offering, Windows Server 2008, includes its first enterpriseworthy virtualization product bundled with the 64-bit version of the OS. Hyper-V, which is a $28 option and is included as a role in the operating system, is the next generation hypervisor-based virtualization technology. It allows you to make the best use of your hardware investments by consolidating multiple server roles as several separate virtual machines running on a single physical machine. Hyper-V will efficiently support multiple operating systems – Windows, Linux and others – in parallel on a single physical server to fully leverage the power of 64-bit computing.

Oracle® Database 10g Enterprise Edition is currently one of the most widely deployed database software packages in the world. Since its introduction in 1977, Oracle database software has been ported to a variety of hardware platforms and Operating Systems. While some database vendors only support one platform or offer different database versions on each platform, Oracle offers a consistent product across all major operating systems, including the Windows operating system.

The Windows Server operating system is a Tier 1 fully supported platform for Oracle. Oracle 7 was the first database to ship on Windows in 1994 and today Windows Server is a base development platform for Oracle. Oracle on Windows is widely deployed in the corporate environment. 29% of Oracle’s revenue is derived from the Windows platform. The x64 platform allows unlimited scalability and Large Page support for database instances with large memory requirements. It is a top performing DBMS on the Windows Server platform.

Oracle currently supports Oracle Database 10g R2 on the following editions of Windows Server 2008:

- Windows Web Server 2008 (x86 and x64)
- Windows Server 2008 Standard (x86 and x64)
- Windows Server 2008 Enterprise (x86 and x64)
- Windows Server 2008 Datacenter (x86 and x64)

The purpose of this whitepaper is to demonstrate the performance capabilities of the Hyper-V virtual machine as compared to the underlying physical machine and to highlight some of the advantages and usage scenarios of virtualization. For this performance comparison, two platforms were tested -- one a physical server and the other a Hyper-V virtual machine. Both were configured the same with the Windows Server 2008 Enterprise Edition x64 operating system and Oracle Database 10g R2. A free database testing and demonstration tool, SwingBench v2.3, was used to perform the benchmark tests. The goal of this comparison is to demonstrate that the Hyper-V virtual platform can deliver performance as good or better as can be achieved on a standalone physical machine for the Oracle 10g Database environment.

Testing Procedures

The goal of the testing was to measure and compare the resource usage and the Transactions per Minute/ Second (TPM / TPS) achieved on separate database servers (physical and virtual) when both machines are identically configured.
Note: The two Dell™ PowerEdge™ servers used for this benchmark, Physical01 and Physical02, were configured exactly the same. The underlying physical machine that hosted the HyperV01 virtual machine (Physical02) was configured with 2 Quad-Core processors and 16.0 GB of physical memory. For the HyperV01 virtual machine, 14.0 GB of memory was allocated and 4 virtual processors were defined. The configuration of the Physical01 server was downgraded for available memory and number of CPUs to match that of the HyperV01 virtual machine in order to provide an “apples-to-apples” comparison. Table 1 shows the configuration of the two platforms utilized for the benchmark tests.

<table>
<thead>
<tr>
<th></th>
<th>HyperV01</th>
<th>Physical01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Memory</td>
<td>14.0 GB</td>
<td>14.0 GB</td>
</tr>
<tr>
<td>CPUs</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows 2008 Enterprise Edition x64 SP1</td>
<td>Windows 2008 Enterprise Edition x64 SP1</td>
</tr>
<tr>
<td>Oracle Database Name</td>
<td>DBVM01</td>
<td>DBPHYS</td>
</tr>
<tr>
<td>Oracle Version</td>
<td>10.2.0.4.0</td>
<td>10.2.0.4.0</td>
</tr>
<tr>
<td>Database SGA size</td>
<td>4.0 GB</td>
<td>4.0 GB</td>
</tr>
<tr>
<td>Database PGA size</td>
<td>1.0 GB</td>
<td>1.0 GB</td>
</tr>
<tr>
<td>Windows Large Page Support Configured?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1 – Server Configuration for HyperV01 and Physical01

Two SwingBench benchmark test sets were executed for this comparison.

The full transaction mix consisted of the standard set of Order Entry OLTP tests. This set included the following processes:

- Customer Registration
- Browse Products
- Order Products
- Process Orders
- Browse Orders

The “browse-only” transaction mix consisted of the following processes from the Order Entry OLTP tests:

- Browse Products
- Browse Orders

For both sets of tests, the number of concurrent users was set to 50 and the load ratio was set at 50% for each transaction process of the set. This criteria was utilized for both the DBPHYS (Physical01) and DBVM01 (HyperV01) databases.

Test Results / Conclusions

The results of the two benchmark sets against each machine configuration are described in the following sections. The Windows Resource Monitor was used to measure CPU and Memory utilization, as the SwingBench CPU monitor is designed to work only on the Linux / UNIX platform.
OLTP Test Results

The summary results of the full OLTP tests are shown in Table 2.

<table>
<thead>
<tr>
<th>Application: SwingBench Order Entry - Full OLTP Transaction Mix (4 CPUs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark Scenario</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>HyperV01</td>
</tr>
<tr>
<td>Physical01</td>
</tr>
</tbody>
</table>

Table 2 - SwingBench Full OLTP Test Results on HyperV01

The full OLTP test utilized 95% of CPU and 46% of available memory on the virtual platform (HyperV01) as shown in Figure 1.

Figure 1 - SwingBench Full OLTP Test Resource Overview on HyperV01

Figure 2 presents an overview of the OLTP benchmark on the virtual platform, which includes the average and maximum Transactions per Minute (TPMs) and Transactions per Second (TPSs).

Figure 2 - SwingBench Full OLTP Benchmark Test Overview on HyperV01
Transactions per Minute (TPMs) and Transactions Transaction Response Time for the OLTP benchmark on the virtual platform is shown in Figure 3.
The results achieved on the physical platform (Physica01) for the full OLTP tests reflect slightly better numbers for CPU and Memory utilization and for TPMs and TPSs.

**Browse-Only Test Results**

The summary results of the “Browse-Only” OLTP tests are shown in Table 3.

<table>
<thead>
<tr>
<th>Application</th>
<th>Total Concurrent Users</th>
<th>Avg. CPU Utilization</th>
<th>Storage Utilization</th>
<th>Memory Utilization (4GB SGA)</th>
<th>Swap Utilization</th>
<th>Average TPMs Achieved</th>
<th>Max TPMs Achieved</th>
<th>Average TPSs Achieved</th>
<th>Max TPSs Achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>HyperV01</td>
<td>50</td>
<td>15%</td>
<td>64% / 74%</td>
<td>43%</td>
<td>0%</td>
<td>48,463</td>
<td>51,160</td>
<td>844</td>
<td>876</td>
</tr>
<tr>
<td>Physical01</td>
<td>50</td>
<td>10%</td>
<td>66% / 78%</td>
<td>42%</td>
<td>0%</td>
<td>47,148</td>
<td>52,290</td>
<td>819</td>
<td>891</td>
</tr>
</tbody>
</table>

Table 3 - SwingBench “Browse-Only” Test Results on HyperV01

The “Browse-Only” test utilized only 15% of CPU and 43% of available memory on the virtual platform (HyperV01) as shown in Figure 4.
Figure 5 presents an overview of the “Browse-Only” benchmark on the virtual platform, which includes the average and maximum Transactions per Minute (TPMs) and Transactions per Second (TPSs).
Transactions per Minute (TPMs) and Transaction Response Time for the “Browse-Only” benchmark on the virtual platform is shown in Figure 6.
Once again, the results achieved on the physical platform (Physica01) for the “Browse-Only” tests reflect slightly better numbers for CPU and Memory utilization. However, the numbers for Average TPMs and TPSs are better for the virtual platform when running the “Browse-Only” scenario.

(note that the “Browse-Only benchmark include just the “Browse Products” and “Browse Orders” tests in the Transaction Response Time graph)

Conclusions

From the analysis of the full OLTP tests, the following was observed

- The difference in Average TPMs Achieved between HyperV01 and Physical01 was 1%.
- The difference in Average TPSs Achieved between HyperV01 and Physical01 was 2%.
- The difference in Max TPMs Achieved between HyperV01 and Physical01 was 5%.
- The difference in Max TPSs Achieved between HyperV01 and Physical01 was 9%.

From the analysis of the “Browse-Only” OLTP tests, the following was observed

- The difference in Average TPMs Achieved between HyperV01 and Physical01 was -3%.
- The difference in Average TPSs Achieved between HyperV01 and Physical01 was -3%.
- The difference in Max TPMs Achieved between HyperV01 and Physical01 was 2%.
- The difference in Max TPSs Achieved between HyperV01 and Physical01 was 2%.

The positive percentage differences for the Average TPMs and TPSs achieved indicate a slight advantage for the physical machine platform for the full OLTP tests.

The -3% differences achieved on the Average TPM and TPS for the “Browse-Only” tests show that the averages achieved on HyperV01 were superior to that achieved on Physical01.

Analysis of the overall results achieved lead to the following conclusions.

- The Hyper-V virtual machine Oracle 10g database was able to provide an almost identical performance to the Oracle 10g database running on the physical database server. This clearly demonstrates that a Hyper-V virtual environment could sustain the performance levels required in a multi-server organization while offering a considerable savings in terms of hardware investment.
- Hyper-V guest virtual machines are limited to a maximum of four CPU cores; therefore, you should run Oracle 10g within Hyper-V guest virtual machines only if your workload performance can be satisfied by no more than four CPUs.
- When compared against native configurations with comparable hardware resources, the same throughput can be achieved within a guest virtual machine at a cost of slightly increased CPU utilization. Proper hardware sizing is critical to Oracle 10g performance. You should ensure that cumulative physical CPU resources on a server are adequate to meet the needs of the guest virtual machines by testing your workload in the planned virtualized environment.
- The information captured so far is specific to performance considerations; for your deployment, take functional considerations (i.e., supported configurations, options to achieve high availability, and so on) also into account.
• For consolidation scenarios, the amount of storage resources available as well as the scenario will drive your decision. You should size your storage with your workload and response time requirements in mind. Always follow best practices with respect to the underlying storage in Hyper-V environments, just as you would with any Oracle 10g deployment.

The results achieved in these tests clearly demonstrate that the Oracle 10g Database, running on the Microsoft Server 2008 operating system, will deliver an almost identical performance regardless of whether the underlying platform is a physical server or a Hyper-V virtual machine.

Testing Environment

The testing environment consisted of 2 Dell PowerEdge (PE) 2900 servers connected to a Dell PowerVault™ MD3000i SAN via an iSCSI interface. The MD3000i contained 2 shelves of 15 x 145GB drives. Each shelf was connected to 1 PE 2900 through a redundant set of Dell PowerConnect™ 5424 network switches. One PE 2900 server was configured with a Hyper-V virtual machine while the other server was configured to operate as a stand-alone box. A Dell OptiPlex™ 755 workstation was used to access the Oracle database on each server in a “client-server” configuration.

The lab setup is depicted in Figure 7.

For more information about the hardware that was used for these tests, see Appendix C.
Hyper-V Configuration Considerations

The Hyper-V virtualization environment requires the use of specific hardware.

- Use a server processor that supports hardware-assisted virtualization. There are two to choose from:
  - Intel VT
  - AMD virtualization (AMD-V)
- Ensure that hardware-assisted virtualization and Data Execution Prevention (DEP) are present and enabled. (You can verify this in the BIOS setting.)
- Run the Hyper-V server role on the root partition only of the Windows® operating system.
- Set any disks that will be configured as “pass-through” disks for the guest virtual machine as offline in the root partition using DISKPART or Volume Manager.
- Insure that the integration components (“enlightenments”) are installed on the guest virtual machine.
- Use a physical network adapter instead of an emulated network adapter when configuring networking for the virtual machine.
- Avoid emulated devices when possible. These devices can result in significantly more CPU overhead when compared to physical devices.

Hyper-V Storage Considerations

Hyper-V supports several different types of storage options. Each of the storage options can be attached via an IDE or SCSI controller. For the RAW devices used for Oracle 10g ASM, we used the virtual SCSI controller configuration option. For an I/O intensive application such as Oracle, it is recommended that you limit your choices to the two best performing options:

- Pass-through disk
- Fixed-sized Virtual Hard Disks (VHDs)

Dynamic VHDs are not recommended for performance reasons. This is because for dynamic VHD, the blocks in the disk start as zeroed blocks, but they are not backed by any actual space in the file. Reads from such blocks return a block of zeros. When a block is first written to, the virtualization stack must allocate space within the VHD file for the block and then update the metadata. In addition to this, every time an existing block is referenced, the block mapping must be looked up in the metadata. This increases both the number of disk I/Os for read and write activities and CPU usage. The dynamic growth also requires the server administrator to monitor disk capacity to ensure that there is sufficient disk storage as the storage requirements increase. Fixed-size VHDs can be expanded if needed, but this requires that the guest virtual machine be shut down during the operation.

Both fixed-sized VHD and pass-through storage configurations were used for the VM machine setup for this paper. The fixed-sized VHD was used for the OS and Oracle binary installation. The pass-through disks were configured as RAW devices for use by the Oracle ASM instance. Synthetic SCSI controllers were used for the guest virtual machines.
Hardware Configuration

*Database Servers*

Physical01 was configured as a standalone physical machine.

Physical02 was configured to support a single VM (HyperV01).

- Both servers consisted of the exact same hardware configuration.
- Each server was connected to a single shelf of the PowerVault MD3000i via an iSCSI interface.
- Each shelf of the MD3000i contained 15 x 145GB disk configured as follows:
  - 6 x 145GB (RAID 1+0) -> 2 x 200GB LUNs configured as RAW devices for use by ASM
  - 6 x 145GB (RAID 1+0) -> 2 x 200GB LUNs configured as RAW devices for use by ASM
  - 2 x 145GB (RAID 1) -> configured as follows:

<table>
<thead>
<tr>
<th>Physical01</th>
<th>Physical02</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x 140GB (Apps – OS and Oracle binaries)</td>
<td>1 x 40GB LUN (Apps – OS and Oracle binaries)</td>
</tr>
<tr>
<td></td>
<td>1 x 100GB LUN (VMs – VM definition files)</td>
</tr>
</tbody>
</table>

See Appendix C for the hardware specification for each PowerEdge 2900 database server.

*Client Workstation*

A Dell OptiPlex 755 workstation was utilized as the client workstation. This machine was named OPT-3.

See Appendix C for the hardware specification for the OptiPlex 755 workstation.

Software Configuration

*OS Configuration*

Both the Physical02 and Physical01 servers were installed with the Windows 2008 Server Enterprise Edition x64 SP1 operating system. All current Microsoft OS updates were installed. The Microsoft iSCSI initiator was used to configure access to the 4 x 200GB LUNs presented from the MD3000i SAN.

The OptiPlex 755 workstation was configured with the Microsoft Windows Vista Business Service Pack 1 32-bit operating system. All current Microsoft OS updates were installed.
Hyper V setup

The HyperV01 virtual machine was created on Physical02 using the steps outlined in the Hyper-V Getting Started Guide. Since it was the only VM running on Physical02, HyperV01 was allocated 14.0GB of memory and 4 virtual CPUs. The attributes for the HyperV01 virtual machine are shown in Figure 10.

![Hyper-V configuration attributes for the HyperV01 virtual machine](image)

The IDE hard drive was configured as a fixed-sized VHD from Physical02 and sized at 20GB. The 4 x 200GB LUNs from the MD3000i were configured as pass-through disks from Physical02 to HyperV01. These disks were configured as RAW devices for use by Oracle Automatic Storage Management (ASM).

Database Servers

In order to achieve an “apples – to – apples” comparison between the VM and the physical database server (Physical01), the bcdedit utility was used to alter the configuration of the Windows boot loader on Physical01 to reduce the amount memory available for use by 2GB and to reduce the number of processors available by 4. This made the Physical01 server configuration the same as the HyperV01 server configuration: 14GB memory and 4 processors.
The following commands were used to modify the Windows 2008 Server boot loader:

```
bootedit /set reservesmemory 2048   # hide 2GB of memory from Windows
bootedit /set Physical01 numproc 4  # sets the number of processors to use
bootedit /esum all /v             # show all entries
```

The Windows boot loader configuration for the Physical01 server is shown below:

```
Windows Boot Loader
---------------------
identifier           (35a3b0cb-90a1-11d8-a8b4-0015178cb52d)
device               partition=1:
push                  \Windows\system32\winload.exe
description          Microsoft Windows Server 2008
locale                en-US
machine               \6f52bf-17e6-41b8-9b3-a0a6f72bb5f4
device                partition=2:
systemroot           \Windows
resumeobject         {35a3b0cb-90a1-11d8-a8b4-0015178cb52d}
num                   OptOut
removenmemory        2048
numproc               4
```

**Oracle Configuration**

Oracle Database 10g Release 2 (version 10.2.0.4.0) Enterprise Edition for Microsoft Windows Vista x64 and Microsoft Windows Server 2008 x64 was installed on the Physical01 physical machine and the HyperV01 virtual machine. Each machine had 4 iSCSI devices (LUNs) at 200GB each in size attached from separate shelves of the MD3000i SAN. These devices were configured as RAW devices for use by the ASM instance. The Oracle ASM library driver was used to label each device as an available candidate disk for use by ASM.

The Database Configuration Assistant (DBCA) was used to create the ASM instance and the Oracle 10g database on each machine. The Oracle database on each machine was configured with the same parameter specifications. See Appendix D for a list of parameter specifications used for each database.

Each database was configured for a 4.0 GB SGA with a 1.0 GB PGA target. As such, the Windows Server 2008 OS was configured to support Windows Large Pages. This is a recommended Oracle best practice when the Oracle SGA is sized at 4.0 GB or greater. The steps for configuring Windows Large Page support are described in Appendix E.

Oracle Database 10g Client Release 2 (version 10.2.0.3.0) for Microsoft Windows Vista and Microsoft Windows Server 2008 (32-bit) was installed on the OPT-3 client workstation. Oracle Net Services was configured to use TCP over the default TNS port to access the DBPHYS and DBVM01 databases on each database server.
SwingBench Configuration

The tool used to perform the benchmark tests was SwingBench. SwingBench is a set of benchmark / test / demo Java based programs originally developed by Dominic Giles of the Oracle U.K Database Solutions group for Oracle Real Application Clusters. Although it is not officially supported by Oracle, SwingBench is freely available via download from here.

SwingBench supports three different schemas: Calling Circle, Order Entry, and Sales History. The Calling Circle schema is designed as a simple OLTP test, usually used for single node tests. The Sales History schema is a DSS (Data Warehouse) benchmark tool. The Order Entry schema emulates a classic TPC-C OLTP benchmark, and is designed to work with Oracle Real Application Clusters. The Order Entry schema from SwingBench v2.3 was used as the basis for the benchmark tests performed for this whitepaper.

SwingBench offers three interfaces for testing from a single client machine. The default SwingBench interface is a Java-based utility that presents rich graphics and an interactive interface. The MiniBench interface is a simple interface with minimal graphics that conserves resources on the client machine. The CharBench interface is completely command line oriented and presents output in an xml file.

The Order Entry TPC-C Benchmark Profile consists of 5 functional application areas: New Customer Registration, Order Products, Process Orders, Browse Products, and Browse Orders. Benchmark tests were performed using the full OLTP benchmark and with only the Browse functions of the OLTP benchmark. The default SwingBench interface was used for all benchmark tests.

The SwingBench v2.3 utility was installed on the OPT-3 client machine. In order to support the SwingBench functionality, Java 1.6 was downloaded and installed on the workstation. The SwingBench environment batch file was updated to reflect the locations of the Oracle client and Java software.

Modifications to the swingbenchenv.bat file:

```
:: If not "%ENVNAMES=" echo "ENVNAMES"
:: Set the following to reflect the root directory of your Java installation
set JAVAHOME=C:\Program Files\Java\jre1.6.0_07
set SWINGBENCH=\C:\swingbench\2.3
set ORACLE_HOME=\Oracle\product\10.2.0\client_1
set ORACLE_HOME=\Oracle\product\10.2.0\client_1
set CLASSPATH=\Oracle\Home1\lib\ojdbc14.jar;\Oracle\Home1\library\ojdbc14.jar
:: The following is needed for JRE environments
set CLASSPATH=\Class\path;\Oracle\Home1\bin\ojdbc14.jar;\Oracle\Home1\bin\ojdbc14.jar
```

Once the swingbenchenv.bat file was modified for the OPT-3 server, the SwingBench Order Entry wizard was invoked for each database to create the schema objects needed for the benchmark tests. Refer to the SwingBench documentation for details regarding this setup.
The Order Entry benchmark setup was defined as follows:

<table>
<thead>
<tr>
<th>Order Entry Service</th>
<th>Schema Owner</th>
<th>Connection Type</th>
<th>Number of Users</th>
<th>Customer Count</th>
<th>Order Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBVM01 (virtual machine)</td>
<td>soe</td>
<td>OCI</td>
<td>50</td>
<td>7,194,489</td>
<td>3,580,954</td>
</tr>
<tr>
<td>DBPHYS (physical machine)</td>
<td>soe</td>
<td>OCI</td>
<td>50</td>
<td>7,194,489</td>
<td>3,580,954</td>
</tr>
</tbody>
</table>

The OCI connection type uses internal packages and procedures in the Oracle schema to process the transactions. This puts the load for the transaction processing on the database server and not the client computer driving the load generator.

The DBVM01 service was configured as preferred to the DBVM01 database on HyperV01 (Hyper-V virtual server).

The DBPHYS service was configured as preferred to the DBPHYS database on Physical01 (Win2K8 physical server).

All services are configured for standard TNS connectivity.

For consistency, a Windows batch script was created to drive each SwingBench benchmark run to insure that each benchmark test was performed with a consistent set of criteria. This script will:

- Drop any user schema’s and tablespaces from a prior run
- Launch the Order Entry data load wizards to load data for each schema
- Launch the SwingBench Coordinator
- Launch the SwingBench load generators
- Launch the SwingBench Cluster Overview manager
- ... wait for operator to perform the benchmark
- Drop any user schema’s and tablespaces from a prior run

The SwingBench load generators were executed from the OPT-3 client workstation using TNS connectivity to the database. This allowed us to simulate an actual client-server OLTP operation, as would be seen in an operational production environment.

**Recommendations / Deployment Scenarios**

---

2 Content excerpted from Microsoft Hyper-V Server 2008 Deployment Scenarios
http://www.microsoft.com/servers/hyper-v-server/deployment.mspx
Recommendations

• Utilize either pass-through disks or fixed VHDs for your guest virtual machine storage. These are the best option for performance, and they should provide the best results for Oracle workloads. Dynamic VHDs are not recommended due to performance reasons.
• Avoid using emulated devices and instead ensure that integration components for Hyper-V have been installed and synthetic devices are being used for I/O, network, and so on. Synthetic devices will provide the best performance with lowest amount of CPU overhead.
• For workloads that make heavy use of network resources, refer to the Virtualization and Network sections of the Windows Performance Tuning guide for best practices on optimizing network for your particular configuration. Test the performance with of your workload, as workload characteristics can vary greatly.

The ability to use some of these techniques will depend on the hardware capabilities.

Deployment Scenarios

Hyper-V Server 2008 is a good solution for organizations that want to consolidate information on a single server or who have low utilization infrastructure workloads, departmental applications and simple branch office workloads.

Testing and Development

Hyper-V Server 2008 is especially suited to meet testing and development virtualizations needs. Using virtual machines, development staffs can create and test a wide variety of scenarios in a safe, self-contained environment that accurately approximates the operation of physical servers and clients.

For example:

• A development team can test the latest version of an application on multiple platforms with a variety of virtual hardware capabilities.
• An IT department can use virtual machines to test deployment of new server and client features. This is especially beneficial in an Oracle environment, where patches and / or new releases of the Oracle software can sometimes break an existing application.

Remote Site Virtualization and Consolidation

Remote sites face several challenges like server virtualization, and the need for remote management due to limited or nonexistent local IT departments. Remote infrastructures benefit from server consolidation. Rather than using multiple small servers, each dedicated to a specific function, such as e-mail services, print services, faxing, or vertical applications, those servers can be virtualized on a single mid-level server.
A single-server approach with virtual machines taking over for previously dedicated servers also means reduced power requirements, reduced space requirements, and having only one physical server to manage (from a hardware perspective).

Under certain circumstances, Microsoft Hyper-V Server 2008 provides a flexible and responsive virtualization solution that enables organizations to virtualize their remote site infrastructure. Businesses can consolidate small remote site servers with Hyper-V Server 2008 to virtualize low utilization infrastructure workloads, departmental applications and simple branch office workloads. For example, Hyper-V Server 2008 is a good solution if your organization is consolidating low utilized servers (file, print, or perhaps a local read-only DC for a small branch office).

Hyper-V Server 2008 is particularly suited for remote sites because it is a stand-alone product which leverages the vast set of drivers already built for Windows and provides high performance with less overhead for device virtualization with new IO virtualization architecture. Hyper-V Server 2008 also enables complete scripted control of virtual machine environments through comprehensive WMI interfaces. In addition, Hyper-V Server 2008 also provides remote sites with data protection and backup with its integrated VSS support for backup.

**Repurposing Older Hardware**

Hyper-V Server 2008 is especially suited for organizations that are consolidating and decommissioning older hardware who want to consolidate information to extend the hardware’s life with better functionality. Microsoft Windows Server 2008 with the Hyper-V role enabled can run on legacy 64-bit hardware (with the right type of CPU and BIOS capability – see Appendix B) and Hyper-V will then support legacy operating systems such as Microsoft® Windows Server 2000, Windows Server 2003, and Linux workloads, or workloads virtualized with Microsoft Virtual Server. Therefore, Hyper-V is a good solution for consolidating information on a single stand-alone server of older hardware.

**Appendix A**

**Hyper-V Configuration**

Hyper-V is a hypervisor-based virtualization technology for Windows Server 2008. The hypervisor is the processor-specific virtualization platform that allows multiple isolated operating systems to share a single hardware platform.

Hyper-V supports isolation in terms of a partition. A partition is a logical unit of isolation, supported by the hypervisor, in which operating systems execute. The Microsoft hypervisor must have at least one parent, or root, partition, running Windows Server 2008 64-bit Edition. The virtualization stack runs in the parent partition and has direct access to the hardware devices. The root partition then creates the child partitions which host the guest operating systems. A root partition creates child partitions using the hypercall application programming interface (API).

---

3 The content in this appendix was excerpted from the whitepaper: Running SQL Server 2008 in a Hyper-V Environment - Best Practices and Performance Recommendations October 2008 by Lindsey Allen, Mike Ruthruff, and Prem Mehra
Partitions do not have access to the physical processor, nor do they handle the processor interrupts. Instead, they have a virtual view of the processor and run in a virtual memory address region that is private to each guest partition. The hypervisor handles the interrupts to the processor, and redirects them to the respective partition. Hyper-V can also hardware accelerate the address translation between various guest virtual address spaces by using an Input Output Memory Management Unit (IOMMU) which operates independent of the memory management hardware used by the CPU. An IOMMU is used to remap physical memory addresses to the addresses that are used by the child partitions.

Child partitions do not have direct access to other hardware resources and are presented a virtual view of the resources, as virtual devices (VDevs). Requests to the virtual devices are redirected either via the VMBus or the hypervisor to the devices in the parent partition, which handles the requests. The VMBus is a logical inter-partition communication channel. The parent partition hosts Virtualization Service Providers (VSPs) which communicate over the VMBus to handle device access requests from child partitions. Child partitions host Virtualization Service Consumers (VSCs) which redirect device requests to VSPs in the parent partition via the VMBus. This entire process is transparent to the guest operating system.

Virtual Devices can also take advantage of a Windows Server Virtualization feature, named Enlightened IO, for storage, networking, graphics, and input subsystems. Enlightened IO is a specialized virtualization-aware implementation of high level communication protocols (such as SCSI) that utilize the VMBus directly, bypassing any device emulation layer. This makes the communication more efficient but requires an enlightened guest that is hypervisor and VMBus aware. Hyper-V enlightened I/O and a hypervisor aware kernel is provided via installation of Hyper-V integration services. Integration components, which include virtual server client (VSC) drivers, are also available for other client operating systems. Hyper-V requires a processor that includes hardware assisted virtualization, such as is provided with Intel VT or AMD Virtualization (AMD-V) technology.

Figure 11 provides a high-level overview of the architecture of a Hyper-V environment running on Windows Server 2008.
Acronyms and terms used in the diagram above are described below:

- **APIC** – Advanced Programmable Interrupt Controller – A device which allows priority levels to be assigned to its interrupt outputs.
- **Child Partition** – Partition that hosts a guest operating system - All access to physical memory and devices by a child partition is provided via the Virtual Machine Bus (VMBus) or the hypervisor.
- **Hypercall** – Interface for communication with the hypervisor - The hypercall interface accommodates access to the optimizations provided by the hypervisor.
- **Hypervisor** – A layer of software that sits between the hardware and one or more operating systems. Its primary job is to provide isolated execution environments called partitions. The hypervisor controls and arbitrates access to the underlying hardware.
- **IC** – Integration component – Component that allows child partitions to communicate with other partitions and the hypervisor.
- **I/O stack** – Input/output stack
- **MSR** – Memory Service Routine
- **Root Partition** – Manages machine-level functions such as device drivers, power management, and device hot addition/removal. The root (or parent) partition is the only partition that has direct access to physical memory and devices.
- **VID** – Virtualization Infrastructure Driver – Provides partition management services, virtual processor management services, and memory management services for partitions.
- **VMBus** – Channel-based communication mechanism used for inter-partition communication and device enumeration on systems with multiple active virtualized partitions. The VMBus is installed with Hyper-V Integration Services.
- **VMMS** – Virtual Machine Management Service – Responsible for managing the state of all virtual machines in child partitions.
- **VMWP** – Virtual Machine Worker Process – A user mode component of the virtualization stack. The worker process provides virtual machine management services from the Windows Server 2008 instance in the parent partition to the guest operating systems in the child partitions. The Virtual Machine Management Service spawns a separate worker process for each running virtual machine.
- **VSC** – Virtualization Service Client – A synthetic device instance that resides in a child partition. VSCs utilize hardware resources that are provided by Virtualization Service Providers (VSPs) in the parent partition. They communicate with the corresponding VSPs in the parent partition over the VMBus to satisfy a child partitions device I/O requests.
- **VSP** – Virtualization Service Provider – Resides in the root partition and provide synthetic device support to child partitions over the Virtual Machine Bus (VMBus).
- **WinHv** – Windows Hypervisor Interface Library - WinHv is essentially a bridge between a partitioned operating system’s drivers and the hypervisor which allows drivers to call the hypervisor using standard Windows calling conventions
- **WMI** – The Virtual Machine Management Service exposes a set of Windows Management Instrumentation (WMI)-based APIs for managing and controlling virtual machines.
Appendix B

Hardware Requirements

Hyper-V requires specific hardware. You can identify systems that support the x64 architecture and Hyper-V by searching the Windows Server catalog for Hyper-V as an additional qualification (see http://go.microsoft.com/fwlink/?LinkId=111228).

To install and use the Hyper-V role, you will need the following:

- Hardware-assisted virtualization. This is available in processors that include a virtualization option—specifically processors with Intel Virtualization Technology (Intel VT) or AMD Virtualization (AMD-V) technology.
- Hardware-enforced Data Execution Prevention (DEP) must be available and enabled. Specifically, you must enable Intel XD bit (execute disable bit) or AMD NX bit (no execute bit).

Tip:
The settings for hardware-assisted virtualization and hardware-enforced DEP are available in the BIOS. However, the names of the settings may differ from the names identified above. For more information about whether a specific processor model supports Hyper-V, check with the manufacturer of the computer. If you modify the settings for hardware-assisted virtualization or hardware-enforced DEP, you may need to turn off the power to the computer and then turn it back on. Restarting the computer may not apply the changes to the settings.

Memory

The maximum amount of memory that can be used is determined by the operating system, as follows:

- For Windows Server 2008 Enterprise and Windows Server 2008 Datacenter, the physical computer can be configured with up to 1 TB of physical memory, and virtual machines that run either of those editions can be configured with up to 64 GB of memory per virtual machine.
- For Windows Server 2008 Standard, the physical computer can be configured with up to 32 GB of physical memory, and virtual machines that run either of those editions can be configured with up to 31 GB of memory per virtual machine.
**Processors**

Hyper-V is supported on physical computers with up to 24 logical processors. A logical processor can be a core processor or a processor using hyper-threading technology. You can configure up to 4 virtual processors on a virtual machine. However, the number of virtual processors supported by a guest operating system might be lower.

The following are some examples of supported systems and the number of logical processors they provide:

- A single-processor/dual-core system provides 2 logical processors.
- A single-processor/quad-core system provides 4 logical processors.
- A dual-processor/dual-core system provides 4 logical processors.
- A dual-processor/quad-core system provides 8 logical processors.
- A quad-processor/dual-core system provides 8 logical processors.
- A quad-processor/dual-core, hyper-threaded system provides 16 logical processors.
- A quad-processor/quad-core system provides 16 logical processors.

Before proceeding further, it is necessary to upgrade the Oracle Universal Installer (OUI) to version 9.2.0.5. Then you may install OCFS binaries into the Oracle Home directory.

**Networking**

Hyper-V provides the following networking support:

- Each virtual machine can be configured with up to 12 virtual network adapters—8 can be the “network adapter” type and 4 can be the “legacy network adapter” type. The network adapter type provides better performance and requires a virtual machine driver that is included in the integration services packages.
- Each virtual network adapter can be configured with either a static or dynamic MAC address.
- Each virtual network adapter offers integrated virtual local area network (VLAN) support and can be assigned a unique VLAN channel.
- You can have an unlimited number of virtual networks with an unlimited number of virtual machines per virtual network.

*Note:* You cannot connect a virtual network to a wireless network adapter. As a result, you cannot provide wireless networking capabilities to virtual machines.
Storage

Hyper-V supports a variety of storage options. You can use the following types of physical storage with a server that runs Hyper-V:

- Direct-attached storage: You can use Serial Advanced Technology Attachment (SATA), external Serial Advanced Technology Attachment (eSATA), Parallel Advanced Technology Attachment (PATA), Serial Attached SCSI (SAS), SCSI, USB, and Firewire.
- Storage area networks (SANs): You can use Internet SCSI (iSCSI), Fibre Channel, and SAS technologies.
- Network-attached storage You can configure a virtual machine to use the following types of virtual storage.
  - Virtual hard disks of up to 2040 GB. You can use fixed virtual hard disks, dynamically expanding virtual hard disks, and differencing disks.
  - Virtual IDE devices. Each virtual machine supports up to 4 IDE devices. The startup disk (sometimes referred to as the boot disk) must be attached to one of the IDE devices. The startup disk can be either a virtual hard disk or a physical disk.
  - Virtual SCSI devices. Each virtual machine supports up to 4 virtual SCSI controllers, and each controller supports up to 64 disks. This means that each virtual machine can be configured with as many as 256 virtual SCSI disks.
  - Physical disks. Physical disks attached directly to a virtual machine (sometimes referred to as pass-through disks) have no size limitation other than what is supported by the guest operating system.
  - Virtual machine storage capacity. Using virtual hard disks, each virtual machine supports up to 512 TB of storage. Using physical disks, this number is even greater depending on what is supported by the guest operating system.
  - Virtual machine snapshots. Hyper-V supports up to 50 snapshots per virtual machine.

Note: Although a virtual machine must use a virtual IDE device as the startup disk to start the guest operating system, you have many options to choose from when selecting the physical device that will provide the storage for the virtual IDE device. For example, you can use any of the types of physical storage identified in the preceding list.
Appendix C

Hardware Configuration

A description of the hardware used for this whitepaper is presented in Table 4.

<table>
<thead>
<tr>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell PowerEdge 2900 (2 servers)</td>
</tr>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>Cache</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>HBA</td>
</tr>
<tr>
<td>OS</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Storage</td>
</tr>
<tr>
<td>Software</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell PowerVault MD3000i</td>
</tr>
<tr>
<td>Shelves</td>
</tr>
<tr>
<td>Data</td>
</tr>
<tr>
<td>Archive</td>
</tr>
<tr>
<td>OS</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell OptiPlex 755</td>
</tr>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>Cache</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>OS</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Storage</td>
</tr>
<tr>
<td>Software</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Table 4 - Oracle 10g Hyper-V Lab Hardware Configuration

The operating system was configured the same way on both the PE 2900s (Physical01 and Physical02). A single virtual machine (HyperV01) was created on Physical02. The Memory and CPU configuration of the Physical01 physical machine was adjusted to match the Memory and CPU configuration of HyperV01.
Appendix D

Oracle Database Configuration

The Oracle databases on both the Physical01 physical machine and the HyperV01 virtual machine were configured with a 4.0 GB SGA (System Global Area).

Table 5 lists the non-default parameter values for the DBPHYS database on the Physical01 physical server.

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>audit_file_dest</td>
<td>F:\ORACLE\PRODUCT\10.2.0\ADMIN\DBPHYS\ADUMP</td>
</tr>
<tr>
<td>background_dump_dest</td>
<td>F:\ORACLE\PRODUCT\10.2.0\ADMIN\DBPHYS\BDUMP</td>
</tr>
<tr>
<td>compatible</td>
<td>10.2.0.3.0</td>
</tr>
<tr>
<td>control_files</td>
<td>*DATADG/dbhpv2/controlfile/current.256.667653675, *ARCHIVEDG/dbhpv2/controlfile/current.256.667653675</td>
</tr>
<tr>
<td>core_dump_dest</td>
<td>F:\ORACLE\PRODUCT\10.2.0\ADMIN\DBPHYS\CDUMP</td>
</tr>
<tr>
<td>db_block_size</td>
<td>5120</td>
</tr>
<tr>
<td>db_create_file_dest</td>
<td>*DATADG</td>
</tr>
<tr>
<td>db_create_online_log_dest_1</td>
<td>*DATADG</td>
</tr>
<tr>
<td>db_create_online_log_dest_2</td>
<td>*ARCHIVEDG</td>
</tr>
<tr>
<td>db_domain</td>
<td>LB02.local</td>
</tr>
<tr>
<td>db_file_multiblock_read_count</td>
<td>16</td>
</tr>
<tr>
<td>db_name</td>
<td>DBPHYS</td>
</tr>
<tr>
<td>db_recovery_file_dest</td>
<td>*ARCHIVEDG</td>
</tr>
<tr>
<td>db_recovery_file_dest_size</td>
<td>400G</td>
</tr>
<tr>
<td>dispatchers</td>
<td>(PROTOCOL=TCP) (SERVICE=DBPHYSXDS)</td>
</tr>
<tr>
<td>job_queue_processes</td>
<td>10</td>
</tr>
<tr>
<td>log_archive_dest_1</td>
<td>LOCATION=ARCHIVEDG/</td>
</tr>
<tr>
<td>log_archive_format</td>
<td>ARCHIVE_RGB.9T</td>
</tr>
<tr>
<td>open_currsors</td>
<td>300</td>
</tr>
<tr>
<td>pga_aggregate_target</td>
<td>1G</td>
</tr>
<tr>
<td>processes</td>
<td>600</td>
</tr>
<tr>
<td>remote_login_passwordfile</td>
<td>EXCLUSIVE</td>
</tr>
<tr>
<td>sessions</td>
<td>665</td>
</tr>
<tr>
<td>sga_target</td>
<td>40</td>
</tr>
<tr>
<td>spfile</td>
<td>*DATADG/dbhpv2/spfile/dbhpv2.ora</td>
</tr>
<tr>
<td>undo_management</td>
<td>AUTO</td>
</tr>
<tr>
<td>undo_tablespace</td>
<td>UNDOTBS1</td>
</tr>
<tr>
<td>user_dump_dest</td>
<td>F:\ORACLE\PRODUCT\10.2.0\ADMIN\DBPHYS\UDUMP</td>
</tr>
</tbody>
</table>

Table 5 - DBPHYS database parameters on physical machine Physical01

Table 6 lists the non-default parameter values for the DBVM01 database on the HyperV01 virtual server.
Both databases were configured with exactly the same settings. Both Window Server 2008 operating systems were configured to support Windows Large Pages.

Appendix E

How to Configure the Windows OS for Large Page Support

As with Linux, large pages can be allocated in Windows. Unlike Linux where you allocate the pages in the OS, with Windows, you configure Oracle or SQL Server to use the large pages and they allocated them.

Large Page Support

Large page support is a feature of Oracle Database 10g Release 1 (10.1) or later. It provides a performance boost for memory-intensive database instances. By taking advantage of newly introduced operating system support, Oracle Database 10g Release 1 (10.1) or later can now make more efficient use of processor memory addressing resources. Specifically, when large page support is enabled, the CPUs in the system will be able to access the Oracle Database buffers in RAM more quickly. Instead of addressing the buffers in 4KB increments, the CPUs are told to use 4MB page sizes when addressing the database buffers.
This feature is particularly useful when the Oracle buffer cache is several gigabytes. Smaller-sized configurations will still see a gain when using large pages, but the gain will not be as great as when the database is accessing large amounts of memory.

To enable large page support, the system administrator must grant the “Lock pages in memory” privilege to the oracle user. This privilege is not enabled by default when Windows is installed.

To grant this privilege:

| Choose Start > Settings > Control Panel. | The Control Panel window opens. |
| Double-click Local Security Policy. | The Local Security Settings window opens. |
| In the left pane of the Local Security Settings window, expand Local Policies and select User Rights Assignment. | |
| In the right pane of the Local Security Settings window, select Lock pages in memory and choose Action > Security... | The Local Security Policy Setting dialog opens. |
| Click Add... | The Select Users or Groups dialog opens. |
| Select the oracle user from the Name list. | |
| Click Add. | |
| Click OK to close the Select Users or Groups dialog. | |
| Click OK to close the Local Security Policy Setting dialog. | |

To take advantage of large pages on Windows Server 2003/2008, the amount of physical memory must be greater than the amount of System Global Area (SGA) specified in the parameter file.

Large pages may not be allocated at all times during instance startup; either all of the SGA is allocated using large pages or all of SGA is allocated using regular pages.

**Note:** Large page usage locks the entire SGA into physical memory. Physical memory is not released during a shrink operation.

**See Also:** Your operating system documentation for restrictions on allocating large pages.
To enable large page support:

1. Go to `ORACLE_BASE\ORACLE_HOME\bin\oracle.key`.
2. Open the file `oracle.key` in a text editor and record the value found.
   It is set by the Oracle Universal Installer. The default is: `SOFTWARE\ORACLE\KEY_HOME_NAME`
   Ex: `SOFTWARE\ORACLE\KEY_OraDb10g_home1`
3. Start Registry Editor at the command prompt:
   `C:\> regedit SOFTWARE\ORACLE\KEY_OraDb10g_home1`

**Note**: Although the Registry Editor lets you view and modify registry keys and parameter values, you normally are not required to do so. In fact, you can render your system useless if you make incorrect changes. Therefore, only advanced users should edit the registry!

Back up your system before making any changes in the registry.

4. Go to `HKEY_LOCAL_MACHINE`.
   Find the key corresponding to the value found in `oracle.key`.
   In the default case, for example, you would look for:
   `HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_HOME_NAME`
5. Create one of the following, depending on where you want to enable large page support:
   (new string value)
   - `ORA_LPENABLE=1` to enable large page support on all instances
   - `ORA_SID_LPENABLE=1` to enable large page support on a specific instance
6. Exit the Registry Editor.
   By default Oracle allocates the minimum available large page size when using large pages. The minimum available large page size, 16 MB, is obtained by using the `GetLargePageMinimum` function.

**Caution**: Do not set the initialization parameter `lock_sga` when large pages are enabled. Large page usage locks the entire SGA into physical memory. When used with parameter `lock_sga`, database startup fails with an error because the operating system automatically locks. That is, it prevents memory from being paged to disk when large pages are requested. Physical memory is not released during a shrink operation.
References

Windows Server 2008 Product Overview

Hyper-V Getting Started Guide

Leverage Virtualization

Virtualization and Consolidation with Hyper-V

Microsoft Hyper-V Server 2008 Deployment Scenarios
http://www.microsoft.com/servers/hyper-v-server/deployment.mspx

Performance Tuning Guidelines for Windows Server 2008
http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.mspx

Hyper-V: Under the Hood May 1, 2008 ∙ by Chris Wolf

Which Windows Server Virtualization 08 storage is best for you?

Storage options for Windows Server 2008 Hyper-V
http://blogs.technet.com/josebda/archive/2008/02/14/storage-options-for-windows-server-2008-shyper-v.aspx

Microsoft Hyper-V + QLogic HBAs = Performance

Running SQL Server 2008 in a Hyper-V Environment - Best Practices and Performance Recommendations
October 2008 by Lindsey Allen, Mike Ruthruff, and Prem Mehra

A Comparison of Oracle Database 10g R2 Real Application Cluster Performance on Microsoft Windows 2003 Server Enterprise Edition x64 and Red Hat Enterprise Linux x86-64
October 2006 by Larry Pedigo, Performance Tuning Corporation.