The Scalable Enterprise: VMware ESX Server on the Dell PowerEdge 6650

Enterprise Product Group (EPG)

Dell White Paper

By Dave Jaffe, Todd Muirhead and Felipe Payet

Dell Enterprise Systems

February 2004
Contents

Executive Summary .............................................................................................................. 4
Introduction ................................................................................................................... 5
  Dell & VMware .................................................................................................................... 5
  Dell’s Scalable Enterprise Vision and Server Virtualization ...................................... 6
  The Demonstration .............................................................................................................. 6
The Hardware ................................................................................................................... 7
VMware Setup ................................................................................................................... 10
  ESX Server 2.0.1 .................................................................................................................. 10
  VirtualCenter ...................................................................................................................... 11
  Virtual Machines .................................................................................................................. 11
The Database Application ................................................................................................... 12
  The Database Schema ....................................................................................................... 12
  The Stored Procedures ...................................................................................................... 13
  The Driver Applications .................................................................................................... 13
    Online Transaction Processing ......................................................................................... 13
    Reports ............................................................................................................................... 14
VMotion in Action .............................................................................................................. 15
Conclusions ....................................................................................................................... 19
Acknowledgements ............................................................................................................ 20
Appendix A. DS Database Build Script ............................................................................. 21
Appendix B. DS Stored Procedures ................................................................................... 24

Tables
Table 1: Details of PowerEdge 6650s used in VMware software test ................................................... 7
Table 2: Dell/EMC Storage Configuration ..................................................................................... 8
Table 3: Dell/EMC Storage Layout ............................................................................................ 8
Table 4: VMware Virtual Machine Settings ............................................................................... 11
Table 5 DVD Store Database Schema ...................................................................................... 12
Table 6: OLTP Driver Parameters................................................................................................................. 14
Table 7: Results of VMotion migration of SQL Server virtual machine while under stress........... 16

Figures
Figure 1: Configuration of Servers and Storage Involved in VMware Software Testing........... 9
Figure 2: VMware VirtualCenter.................................................................................................................. 15
Figure 3: VMotion migration of SQL Server Database.............................................................................. 17
Figure 4: ESX Server Service Console ....................................................................................................... 18
Executive Summary

This paper introduces the server virtualization software, VMware® ESX Server™ 2.0.1, VirtualCenter and VMotion™, on Dell™ PowerEdge™ servers. Server virtualization can enable cost and space savings through server consolidation and load balancing, rapid standardized deployment and enhanced manageability. This paper discusses these features, followed by a detailed discussion of how ESX Server and VirtualCenter were installed on Dell servers using a Dell/EMC® Storage Area Network (SAN). Finally, the benefits of running VMware software on clusters of 4-way servers were demonstrated by moving an online store database application, running under SQL Server 2000 Enterprise Edition on a VMware virtual machine, from one Dell PowerEdge 6650 server to another under load (100 orders per second), without losing any transactions, and with a barely noticeable rise in response time as seen by the end user.
Introduction

Dell & VMware

Dell sees IT managers today facing a number of challenges as they are increasingly pushed to “do more with less”, improving service delivery while lowering operational costs:

• Decrease “server sprawl”
• Improve system utilization
• Standardize systems
• Reduce system management complexity
• Improve flexibility of IT resources

As part of Dell’s strategy for the Scalable Enterprise, and working in partnership with VMware, Inc., a leader in the emerging virtual infrastructure marketplace, Dell offers a series of server virtualization solutions aimed at addressing these challenges, based on Dell PowerEdge servers, VMware ESX Server virtualization software, VirtualCenter virtual machine management, and VMotion virtual machine migration technology.

VMware ESX Server software is used for partitioning and consolidating systems. When deployed on Dell PowerEdge servers, ESX Server allows the user to create multiple “virtual machines” on a single physical server, each of which is then able to run a separate operating system and applications, without interfering with other virtual machines on the physical server.

VMware VirtualCenter management software provides a central point of control for the data center’s computing resources. It is a cost-effective, highly scalable virtual machine monitoring and management platform with advanced resource management capabilities.

VMotion technology enables the live migration of a running virtual machine to a different physical server without service interruption, allowing for fast reconfiguration and optimization of resources with minimal user impact.

Dell’s server virtualization solutions are targeted towards specific workload deployments where they offer the most value:

• Test and development environments: VMware virtualization software running on Dell PowerEdge servers can help a customer to consolidate multiple test and development servers onto fewer physical machines without sacrificing flexibility or functionality.
• Application consolidation: As a result of the paradigm of “One Application/One Server” common in server environments, many customers today are concerned about underutilized servers - those dedicated to non-mission-critical applications that are either run infrequently or when run, require only a small percentage of the server’s capacity. VMware virtualization software on Dell servers allows customers to consolidate multiple
legacy or underutilized applications onto fewer physical machines, with the goal of lowering complexity and TCO without compromising stability or security.

**Dell’s Scalable Enterprise Vision and Server Virtualization**

Dell sees the IT market gravitating towards two models – a proprietary model and a standards-based model. Dell, through its singular focus on the needs of the customer, advocates the standards-based model, which focuses on scalable, flexible solutions based upon industry standards and delivering excellent customer value as a result of economies of scale, high-volume production and efficient supply-chain management.

At the core of Dell’s Scalable Enterprise strategy is the belief that smaller, industry-standard building blocks that leverage the volume economics of our industry in Clusters or Grids are vastly preferable to large, expensive, proprietary systems. In the server virtualization/consolidation space, Dell recommends deployments of VMware ESX Server built on multiple 4-way or 2-way industry-standard server building blocks, instead of larger proprietary SMP systems. By deploying VMware ESX Server on multiple servers and leveraging the new VMware VMotion virtual-machine migration technology, Dell’s solution aims to deliver multiple benefits not available on single, larger SMP system-based VMware deployments:

- **Risk mitigation:** Virtual machines distributed among multiple smaller servers mitigate the impact of a hardware failure, compared to a deployment on a single large (8-way or larger) SMP system. When one of a group of servers fails, only that server’s virtual machines are affected. In the single large system scenario, a server failure would affect all of the virtual machines hosted by that server.
- **Expansion flexibility:** A VMware deployment based on smaller building blocks permits a more granular “pay as you grow” approach to expandability, whereby the user can add server capacity via smaller 2- and 4-way servers, as opposed to 8-way and larger systems.
- **Operational flexibility:** A VMware deployment based on multiple Dell PowerEdge 6650 servers and shared Dell/EMC storage, in conjunction with VMware VMotion technology, allows the user to migrate virtual machines live, in real time, from one physical server to another. As we will describe in this document, virtual machines can be moved from server to server to respond to changes in workload demand, as well as to permit hardware upgrades or maintenance, with minimal impact to workload delivery.

**The Demonstration**

To illustrate the benefits of VMware software on clusters of 4-way servers, an online store database application was built on two SQL Server 2000 Enterprise Edition instances deployed as virtual machines on two Dell PowerEdge 6650 servers running VMware ESX Server 2.0.1, with one SQL Server instance receiving orders and a second generating financial reports based on that data. To test whether production virtual machines can be moved from one physical server to another with minimal impact, the virtual machine hosting the order-entry SQL Server database was moved from one physical server to the other, while under load (100 orders per second), without losing any transactions, and with a barely noticeable rise in response time as seen by the end user.
The Hardware

Two Dell PowerEdge 6650 Servers were used for the testing. Each PowerEdge 6650 was loaded with four 2.8 GHz Intel® Xeon™ Processors MP and 4 GB of memory. The two onboard Gigabit Ethernet1 Network Interface Controllers (NICs) were used in addition to an Intel Pro 1000XT Gigabit NIC. Three NICs were used so that the ESX Server Service Console, the virtual machines, and the VMotion feature could each have their own dedicated bandwidth. The details are shown in Table 1.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Each Dell PowerEdge 6650</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware ESX Server 2.0.1</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>4x 2.8 GHz Intel Xeon Processors MP w/ 2MB L3 cache</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB</td>
</tr>
<tr>
<td>Internal Disks</td>
<td>2x 18 GB</td>
</tr>
</tbody>
</table>
| NICs             | 2x 10/100/1000 Mb/s (internal)  
                   | 1 Intel Pro 1000XT Gb           |
| Disk Controller  | PERC/3 Dual Channel      |
| Fibre Channel Host Bus Adapter | QLogic 2340 |
| Height           | 4 Rack Units (4U) or 7-inches |

Table 1: Details of PowerEdge 6650s used in VMware software test

The PowerEdge 6650 servers were attached to a Storage Area Network via the QLogic 2340 fibre channel Host Bus Adapters (HBA). A Dell/EMC CX600 was attached to the SAN to provide storage. We assigned 38 of the 150 drives attached to our CX600 for our VMware virtualization environment. The tables below describe and depict our configuration. Table 2 gives the basic configuration and Table 3 provides a look at how the CX600 storage was organized into LUNs. One LUN was used to stage the data that was loaded into the database. We used the snapshot capability of the EMC SnapView™ software to create a second copy that could be used by the other virtual machine so that multiple systems could load data at the same time.

---

1. Gigabit Ethernet: A networking technology that provides high-speed data transfer rates, typically ranging from 100 to 1000 Mbps. It is commonly used for local area networks (LANs) and is a standard network type that supports Gigabit Ethernet. It is used to connect computer systems, servers, storage devices, and other network elements over a high-speed network.
Controller | 1 Dell/EMC CX600
---|---
Disk Enclosures | 4 Dell/EMC DAE2
Disks | 38 x 73 GB/ 10K RPM
LUNs | 1 6-Disk RAID 5 LUN for Virtual Machine Boot Drive
   | 4 5-Disk RAID 5 LUNs for Database Data Storage
   | 2 2-Disk RAID 1 LUNs for Database Logs
   | 1 5-Disk RAID 5 for Temporary Data Staging for Loading
   | 1 2-Disk RAID 1 for SnapView Cache (Split into two LUNs)
   | 1 HotSpare Disk
Software | Navisphere® Manager
   | Access Logix™
   | SnapView

Table 2: Dell/EMC Storage Configuration

<table>
<thead>
<tr>
<th>Disk Enclosures</th>
<th>Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAE1</td>
<td>RAID 5 – Data</td>
</tr>
<tr>
<td></td>
<td>RAID 5 – virtual machine Boot Drives</td>
</tr>
<tr>
<td>DAE2</td>
<td>RAID 5 – SQL Data1 – VM1</td>
</tr>
<tr>
<td></td>
<td>RAID 5 – SQL Data2 – VM1</td>
</tr>
<tr>
<td></td>
<td>RAID1 – LogsVM1</td>
</tr>
<tr>
<td></td>
<td>RAID 1 - SnapCache</td>
</tr>
<tr>
<td></td>
<td>Hot Spare</td>
</tr>
<tr>
<td>DAE3</td>
<td>RAID5 – SQL Data1 – VM2</td>
</tr>
<tr>
<td></td>
<td>RAID5 – SQL Data2 – VM2</td>
</tr>
<tr>
<td></td>
<td>RAID1 – LogsVM2</td>
</tr>
</tbody>
</table>

Table 3: Dell/EMC Storage Layout

A Dell PowerEdge 2650 was used to drive a load against SQL Server databases that were installed in virtual machines on the servers running ESX Server. All of the servers were connected to a PowerConnect™ 5224 Gigabit Ethernet switch for network connectivity. The two PowerEdge 6650s running ESX Server were additionally connected to the CX600 via a Brocade fibre channel switch. The virtual machines running on the ESX Server installations had all of their storage residing on the SAN, with each virtual machine having a boot drive, two drives for data, and a log drive. When a virtual machine was migrated from one ESX Server to the other, the only thing that actually moved was the contents of that virtual machine’s RAM. All of the storage was already accessible by both ESX Server machines and did not need to move. Figure 1 shows all of the servers involved.
Figure 1: Configuration of Servers and Storage Involved in VMware Software Testing
Section 4

VMware Setup

VMware ESX Server 2.0.1 and VMware VirtualCenter are the two software components that we used from VMware to complete all of our testing. The difference between the two is that ESX Server 2.0.1 is the software that was installed on each of the server nodes and VirtualCenter is a centralized console application that allowed us to manage and control the ESX Server installations and the virtual machines running on them from a single place.

ESX Server 2.0.1

VMware ESX Server contains its own kernel that replaces a typical operating system for the server. ESX Server runs directly on the hardware and acts as a host for virtual machines. This allows for multiple virtual machines to run at the same time on the same hardware; ESX Server manages them all.

The internal drives on the PowerEdge 6650 were set up as a RAID 1 configuration. The QLogic HBA was left disconnected from the SAN during the initial stage of the ESX Server installation.

To install ESX Server we booted from the ESX Server CD-ROM and answered the installation questions concerning partitioning of the local drives, the ESX Server hostname, IP address, DNS Server, Gateway Address, and initial root password. All the files were copied from the install CD-ROM and then the system was rebooted.

To complete installation of the ESX Server software (and for most administration and configuration tasks) a web browser was used to remotely access the Service Console of ESX Server. The first time ESX Server was accessed with the web browser following the initial installation stage, we were led through a series of configuration steps to get the ESX Server installation fully operational. This included installing the ESX Server license and configuring all hardware on the server that would be used by either the Service Console or virtual machines. The Service Console of each ESX Server installation requires a dedicated NIC. It is recommended that the virtual machines also get one or more dedicated NICs that serve all the virtual machines. We set the ownership of the fibre channel HBA, and all of the SAN storage that would be accessible from it, as dedicated to the virtual machines.

Another reboot was required after configuring the ESX Server hardware options. This was a good opportunity to connect the QLogic HBA into the SAN fabric. It was also necessary to create a new zone on the switch for the newly connected server. Once the switch was correctly zoned, we used the management tool for the Dell/EMC SAN, Navisphere Manager, and manually registered the new host in the Connectivity Status screen. This was necessary because there is not currently a version of Navisphere Agent available for ESX Server which would allow the registration to happen automatically. Once the registration had been completed, we used Navisphere Manager to create the necessary
RAID Groups, LUNs, and Storage Groups. All of the ESX Server machines that were to participate in VMotion migrations were assigned to the same Storage Group.

VirtualCenter

VirtualCenter, a Windows based program, was installed on a separate PowerEdge 1750 server running Microsoft® Windows® 2003 Server Enterprise Edition that served as a management node of our VMware virtualization configuration. We decided to install both the client and server portions of VirtualCenter on our PowerEdge 1750 as well as create a database on the server to be used by VirtualCenter. Installation of VirtualCenter on the PowerEdge 1750 was straightforward and completed in a few minutes.

All of the ESX Server machines that were to be managed by VirtualCenter had to be added to the VirtualCenter console. A simple Connect Host wizard prompted for the hostname, user id and password of the ESX Server machine being added. VirtualCenter then contacted the ESX Server machine to be added and made it a part of VirtualCenter. By having all of the ESX Server installations in one VirtualCenter, all of those ESX Server installations’ virtual machines were also manageable from VirtualCenter. This meant that all of these virtual machines could now be included in the VirtualCenter features of cloning, templates, and migration or VMotion virtual machine migration.

Virtual Machines

In order to create our virtual machines, we used VirtualCenter to create a new virtual machine with a 10 GB hard drive to be created on the SAN, 2 CPUs (using the Virtual SMP feature of ESX Server), 1 GB of RAM, and we specified that the operating system for the virtual machine would be Windows 2003 Server Enterprise Edition. VMware ESX Server then created the virtual machine that was ready for installation of the specified operating system. We started up the virtual machine and had it boot from the ISO image of our Windows 2003 Server Enterprise Edition CD-ROM and did an installation in the virtual machine. Following that installation we then installed SQL Server 2000 Enterprise Edition and the necessary SQL Server service packs. We then created two clones of this original virtual machine and used these for our testing. Any additional SQL Server 2003 virtual machines could be quickly created by simply cloning the master again.

After the initial creation of the virtual machines, each was assigned additional hard disks for the data and logs of the database that reside on the CX600 as shown in the previous section.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPUs</td>
<td>2</td>
</tr>
<tr>
<td>RAM</td>
<td>1024 MB</td>
</tr>
<tr>
<td>Hard Disk Size</td>
<td>10 GB</td>
</tr>
</tbody>
</table>

Table 4: VMware Virtual Machine Settings
The Database Application

To demonstrate the advantages of running a large application under VMware ESX Server, a large (100 GB) online DVD store was implemented as two replicated SQL Server 2000 Enterprise Edition databases, each running in its own virtual machine. One of the SQL Server instances handled the entry of new orders and replicated changes on a scheduled basis to the second SQL Server instance, which was used for generating financial reports. The DVD Store (DS) database consisted of a set of data tables organized according to a certain schema, as well as a set of stored procedures that did the actual work of managing the data in the database as orders were entered and reports were requested. The backend of the database was designed to be driven from a Web-based middle tier, but since the focus of Dell’s testing was on the database servers, the backend stored procedures were driven directly by custom C programs to simulate a Web-based middle tier.

The Database Schema

The DVD store was comprised of four main tables and one other (see Table 5). The Customers table was pre-populated with two hundred million customers, with one hundred million US customers and one hundred million customers from the rest of the world. The Orders table was pre-populated with ten million orders per month, starting in January 2003 and ending in September 2003. The Order Lines table was pre-populated with an average of 5 items per order. The Products table contained one million DVD titles. In addition, the Categories table listed the 16 DVD categories. The full DS Database build script is shown in Appendix A.

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
<th>Number of Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>CUSTOMERID, FIRSTNAME, LASTNAME, ADDRESS1, ADDRESS2, CITY, STATE, ZIP, COUNTRY, REGION, EMAIL, PHONE, CREDITCARD, CREDITCARD_EXPIRATION, USERNAME, PASSWORD, AGE, INCOME, GENDER</td>
<td>200 million</td>
</tr>
<tr>
<td>Orders</td>
<td>ORDERID, ORDERDATE, CUSTOMERID, NETAMOUNT, TAX, TOTALAMOUNT</td>
<td>90 million</td>
</tr>
<tr>
<td>Orderlines</td>
<td>ORDERLINEID, ORDERID, PROD_ID, QUANTITY, ORDERDATE</td>
<td>450 million</td>
</tr>
<tr>
<td>Products</td>
<td>PROD_ID, CATEGORY, TITLE, ACTOR, PRICE, QUAN_IN_STOCK, SPECIAL</td>
<td>1 million</td>
</tr>
<tr>
<td>Categories</td>
<td>CATEGORY, CATEGORYNAME</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 5 DVD Store Database Schema
The Stored Procedures

The DVD Store database was managed through seven stored procedures. The first two were used during the log-in phase. If the customer was a returning customer, Log in was used to retrieve the customer’s information, in particular the CUSTOMERID. If the customer was a new customer, New_customer was used to create a new row in the Customers table with the user’s data. Following the log-in phase the customer might search for a DVD by category, actor or title. These were implemented by Browse_by_category, Browse_by_actor and Browse_by_title, respectively. Finally, after the user had made his or her selections, Purchase was used to complete the transaction. Additionally, Rollup_by_category was used to total sales by DVD category for the previous month, quarter and half-year periods. The stored procedures are shown in Appendix B.

The Driver Applications

Separate multi-threaded driver programs were written to model the order entry or OLTP workload and the report request workload.

Online Transaction Processing

Each thread of the OLTP driver application connected to the database and made a series of stored procedure calls that simulated users logging in, browsing and purchasing. Since there were no simulated user think times or key times, the database connections were kept full, simulating what happens in a real multi-tiered application where some small number of connections are pooled and shared among the web servers that may be handling thousands of simultaneous customers. Thus, a very realistic simulation of database activity was achieved without needing to model thousands of users.

Each thread of the driver modeled a series of customers going through the entire sequence of logging in, browsing the catalog several ways and finally purchasing the selected items. Each completed sequence by a customer was counted as a single order. The driver measured order rates and the average response time to complete each order. Several tunable parameters were used to control the application and are described in Table 6.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value(s) used in test</th>
</tr>
</thead>
<tbody>
<tr>
<td>n_threads</td>
<td>Number of simultaneous connections to the database</td>
<td>10</td>
</tr>
<tr>
<td>warmup_time</td>
<td>Warm-up time before statistics are kept</td>
<td>1 min</td>
</tr>
<tr>
<td>run_time</td>
<td>Run time during which statistics are kept</td>
<td>varied</td>
</tr>
<tr>
<td>pct_returning</td>
<td>Percentage of users that are returning users</td>
<td>95%</td>
</tr>
<tr>
<td>pct_new</td>
<td>Percentage of users that are new users</td>
<td>5%</td>
</tr>
<tr>
<td>n_browse_category</td>
<td>Number of searches based on category</td>
<td>Range: 1 - 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average: 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n_browse_actor</td>
<td>Number of searches based on actor</td>
<td>Range: 1 - 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average: 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n_browse_title</td>
<td>Number of searches based on title</td>
<td>Range: 1 - 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average: 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n_line_items</td>
<td>Number of items purchased</td>
<td>Range: 1 – 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average: 5</td>
</tr>
<tr>
<td>net_amount</td>
<td>Total amount of purchase</td>
<td>Range: $0.01 - $400.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average: $200.00</td>
</tr>
</tbody>
</table>

Table 6: OLTP Driver Parameters

**Reports**

The report request driver program was similar to the OLTP driver in that each thread connected to the database and started making stored procedure calls. Each thread made repeated calls to the `Rollup_by_category` stored procedure, which calculates total sales by DVD category for the previous month, quarter and half-year, until all reports for all 16 categories were completed. In each test 8 simultaneous reports were run.
VMotion in Action

To demonstrate the ability to move virtual servers around a farm of physical servers, we used VMotion to move a virtual machine (VM) from one physical server running ESX Server to another while running the SQL Server DVD Store database under a heavy load (100 orders per second). Such a move might be required to load balance or to perform routine maintenance on the server or in response to an alert that a server parameter (for example, the temperature) had exceeded a warning threshold.

The VMware server farm is shown using VirtualCenter in Figure 2.

We started with one node of the SQL Server replication group, w2k3sql3, on physical server ESX6650B, handling approximately 100 orders per second with an average response time of 0.1 sec. (Response time is defined as the total response time seen by the emulated end user for the complete order transaction, including login time, browse time, and the response time after the user presses the submit button to purchase his/her order.)
The second SQL Server system, w2k3sql2, running on ESX6650A, was started up to roll up sales by DVD category for 8 separate categories. In addition, the two servers were set up to replicate new orders from the w2k3sql3 node to the w2k3sql2 node every day at 12:05 AM. After both servers had ramped up to full speed, using about 80% of the two CPUs dedicated to their VMs, the command was given to move the order entry server (w2k3sql3) from physical server ESX6650B to physical server ESX6650A, without stopping either the incoming orders or the rollups on w2k3sql2. The results can be seen in Table 7 and Figure 3. As shown in Figure 3, for the first 25 seconds after the VMotion migration was initiated at 15:36:20, there was little impact on either throughput (orders per second, top half of Figure 3), or response time (bottom half of Figure 3) as VirtualCenter prepared for the move by initializing a new VM on the target ESX Server and synchronizing the memory between the two. At about 15:36:45 the effects of the memory synchronization could be seen in the dropping throughput and increasing response time. Finally, the actual move occurred at 15:37:08 and the response time maxed out at 2.572 seconds while the order handling paused for about two seconds. Immediately after the move the throughput and response time rapidly returned to close to their previous levels. The target ESX Server CPU utilization rose to about 80% as both VMs ran on it using 2 CPUs each. The throughput was down slightly from the pre-VMotion level but was still high enough to handle 300,000 orders per hour while the first system was being repaired or upgraded.

<table>
<thead>
<tr>
<th>Time</th>
<th>New Orders Completed Per Second</th>
<th>Average Response Time (s)</th>
<th>Maximum Response Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before VMotion – VMs Running on Separate Servers</td>
<td>103</td>
<td>0.098</td>
<td>0.201</td>
</tr>
<tr>
<td>During VMotion</td>
<td>80</td>
<td>0.139</td>
<td>2.572</td>
</tr>
<tr>
<td>After VMotion – VMs running on Same Server</td>
<td>93</td>
<td>0.109</td>
<td>0.492</td>
</tr>
</tbody>
</table>

Table 7: Results of VMotion migration of SQL Server virtual machine while under stress
Figure 3: VMotion migration of SQL Server Database.

The VMs are shown after the move in the ESX Server Service Control panel, Figure 4.
Figure 4: ESX Server Service Console
Conclusions

VMware ESX Server 2.0.1 software running on Dell PowerEdge servers with Dell/EMC storage provides a robust platform for server virtualization. In the example described in this paper, two new SQL Server 2000 Enterprise Edition database server virtual machines were rapidly cloned from a single master and then used to implement a large online store with one server handling new orders and replicating them to the second server for reporting. Since the servers are virtual they can be moved from one physical server to another for purposes of load balancing or maintenance. As documented here, a loaded SQL Server virtual machine, handling 100 orders per second, was moved to a second physical server without stopping the application and without losing any transactions. While the entire VMotion migration took 48 seconds, the performance of the database server as seen by the end user was only impacted for the few seconds that the virtual machine was actually switched from one physical server to the other, and the increased response time seen by the user (for the total database server response time including login, browse and purchase) was only a little over 2 seconds.

\(^1\)This term indicates compliance with IEEE standard 802.3ab for Gigabit Ethernet, and does not connote actual operating speed of 1 Gb/sec. For high speed transmission, connection to a Gigabit Ethernet server and network infrastructure is required.

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

Dell, PowerEdge, and PowerConnect are trademarks of Dell Inc. Intel and Xeon are registered trademarks of Intel Corp. VMware, the VMware “boxes” logo, ESX Server, GSX Server, and VMotion are trademarks of VMware, Inc. Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell disclaims proprietary interest in the marks and names of others.

©Copyright 2004 Dell Inc. All rights reserved. Reproduction in any manner whatsoever without the express written permission of Dell Inc. is strictly forbidden. For more information, contact Dell.

Information in this document is subject to change without notice.
The authors would like to thank Craig Lowery, Tim Abels and Wenlong Xu of the Scalable Enterprise Computing team for valuable discussions.
Appendix A. DS Database Build Script

-- Copyright Dell Inc. 2004

-- Database
IF EXISTS (SELECT * FROM SYSDATABASES WHERE NAME='DS')
DROP DATABASE DS
GO

CREATE DATABASE DS ON
  PRIMARY
  (  
    NAME = 'primary',
    FILENAME = 'c:\sql\dbfiles\ds.mdf'
  ),
FILEGROUP DS_MISC_FG
  (  
    NAME = 'ds_misc',
    FILENAME = 'e:\ds_misc.ndf',
    SIZE = 1GB
  ),
FILEGROUP DS_CUST_FG
  (  
    NAME = 'cust1',
    FILENAME = 'e:\cust1.ndf',
    SIZE = 25GB
  ),
  (  
    NAME = 'cust2',
    FILENAME = 'e:\cust2.ndf',
    SIZE = 25GB
  ),
FILEGROUP DS_ORDERS_FG
  (  
    NAME = 'orders1',
    FILENAME = 'e:\orders1.ndf',
    SIZE = 20GB
  ),
  (  
    NAME = 'orders2',
    FILENAME = 'e:\orders2.ndf',
    SIZE = 20GB
  ),
FILEGROUP DS_IND_FG
  (  
    NAME = 'ind1',
    FILENAME = 'e:\ind1.ndf',
    SIZE = 20GB
  ),
  (  
    NAME = 'ind2',
    FILENAME = 'e:\ind2.ndf',
    SIZE = 20GB
  )
LOG ON
  (  
    NAME = 'ds_log',
    FILENAME = 'e:\ds_log.ldf',
    SIZE = 20GB
  )
GO

USE DS
GO

-- Tables
CREATE TABLE CUSTOMERS
  (  
    CUSTOMERID INT IDENTITY NOT NULL,
CREATE TABLE CUSTOMERS
(
FIRSTNAME VARCHAR(50) NOT NULL,
LASTNAME VARCHAR(50) NOT NULL,
ADDRESS1 VARCHAR(50) NOT NULL,
ADDRESS2 VARCHAR(50),
CITY VARCHAR(50) NOT NULL,
STATE VARCHAR(50),
ZIP INT,
COUNTRY VARCHAR(50) NOT NULL,
REGION TINYINT NOT NULL,
EMAIL VARCHAR(50),
PHONE VARCHAR(50),
CREDITCARD VARCHAR(50) NOT NULL,
CREDITCARDEXPIRATION VARCHAR(50) NOT NULL,
USERNAME VARCHAR(50) NOT NULL,
PASSWORD VARCHAR(50) NOT NULL,
AGE TINYINT,
INCOME INT,
GENDER VARCHAR(1)
)
ON DS_CUST_FG
GO

CREATE TABLE ORDERS
(
ORDERID INT IDENTITY NOT NULL,
ORDERDATE DATETIME NOT NULL,
CUSTOMERID INT NOT NULL,
NETAMOUNT MONEY NOT NULL,
TAX MONEY NOT NULL,
TOTALAMOUNT MONEY NOT NULL
)
ON DS_ORDERS_FG
GO

CREATE TABLE ORDERLINES
(
ORDERLINEID SMALLINT NOT NULL,
ORDERID INT NOT NULL,
PROD_ID INT NOT NULL,
QUANTITY SMALLINT NOT NULL,
ORDERDATE DATETIME NOT NULL
)
ON DS_ORDERS_FG
GO

CREATE TABLE PRODUCTS
(
PROD_ID INT IDENTITY NOT NULL,
CATEGORY TINYINT NOT NULL,
TITLE VARCHAR(50) NOT NULL,
ACTOR VARCHAR(50) NOT NULL,
PRICE MONEY NOT NULL,
QUAN_IN_STOCK INT NOT NULL,
SPECIAL TINYINT
)
ON DS_MISC_FG
GO

CREATE TABLE CATEGORIES
(
CATEGORY TINYINT IDENTITY NOT NULL,
CATEGORYNAME VARCHAR(50) NOT NULL,
)
ON DS_MISC_FG
GO

SET IDENTITY_INSERT CATEGORIES ON
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (1,'Action')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (2,'Animation')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (3,'Children')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (4,'Classics')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (5,'Comedy')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (6,'Documentary')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (7,'Drama')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (8,'Family')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (9,'Foreign')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (10,'Games')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (11,'Horror')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (12,'Music')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (13,'New')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (14,'Sci-Fi')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (15,'Sports')
INSERT INTO CATEGORIES (CATEGORY, CATEGORYNAME) VALUES (16,'Travel')
GO
Appendix B. DS Stored Procedures

-- copyright Dell Inc. 2004

-- NEW_CUSTOMER

USE DS
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'NEW_CUSTOMER' AND type = 'P')
    DROP PROCEDURE NEW_CUSTOMER
GO
USE DS
GO

CREATE PROCEDURE NEW_CUSTOMER
    (@firstname_in VARCHAR(50),
    @lastname_in VARCHAR(50),
    @address1_in VARCHAR(50),
    @address2_in VARCHAR(50),
    @city_in VARCHAR(50),
    @state_in VARCHAR(50),
    @zip_in INT,
    @country_in VARCHAR(50),
    @region_in TINYINT,
    @email_in VARCHAR(50),
    @phone_in VARCHAR(50),
    @creditcard_in VARCHAR(50),
    @creditcardexpiration_in VARCHAR(50),
    @username_in VARCHAR(50),
    @password_in VARCHAR(50),
    @age_in TINYINT,
    @income_in INT,
    @gender_in VARCHAR(1))
AS

IF (SELECT COUNT(*) FROM CUSTOMERS WHERE USERNAME=@username_in) = 0
BEGIN
    INSERT INTO CUSTOMERS
    (FIRSTNAME,
    LASTNAME,
    ADDRESS1,
    ADDRESS2,
    CITY,
    STATE,
    ZIP,
    COUNTRY,
    REGION,
    EMAIL,
    PHONE,
    CREDITCARD,
    CREDITCARDEXPIRATION,
    USERNAME,
    PASSWORD,
    AGE,
    INCOME,
    GENDER)
    VALUES
    (@firstname_in,
    @lastname_in,
    @address1_in,
    @address2_in,
    @city_in,
    @state_in,
@zip_in,
@country_in,
@region_in,
@email_in,
@phone_in,
@creditcard_in,
@creditcardexpiration_in,
@username_in,
@password_in,
@age_in,
@income_in,
@gender_in
)
SELECT @@IDENTITY
END
ELSE
SELECT 0
GO

-- LOGIN
USE DS
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'LOGIN' AND type = 'P')
  DROP PROCEDURE LOGIN
GO
USE DS
GO
CREATE PROCEDURE LOGIN
(  
  @username_in          VARCHAR(50),
  @password_in          VARCHAR(50)
)
AS
  DECLARE @customerid_out INT

  SELECT @customerid_out=CUSTOMERID FROM CUSTOMERS WHERE USERNAME=@username_in AND PASSWORD=@password_in
  IF (@@ROWCOUNT > 0)
    SELECT @customerid_out
  ELSE
    SELECT 0
GO
USE DS
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'BROWSE_BY_CATEGORY' AND type = 'P')
  DROP PROCEDURE BROWSE_BY_CATEGORY
GO
USE DS
GO
CREATE PROCEDURE BROWSE_BY_CATEGORY
(  
  @batch_size_in        INT,
  @category_in          INT
)
AS
  SET ROWCOUNT @batch_size_in
  SELECT * FROM PRODUCTS WHERE CATEGORY=@category_in
  SET ROWCOUNT 0
GO
USE DS
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'BROWSE_BY_ACTOR' AND type = 'P')
    DROP PROCEDURE BROWSE_BY_ACTOR
GO
USE DS
GO
CREATE PROCEDURE BROWSE_BY_ACTOR
    (@batch_size_in INT,
    @actor_in VARCHAR(50))
AS
    SET ROWCOUNT @batch_size_in
    SELECT * FROM PRODUCTS WHERE ACTOR=@actor_in
    SET ROWCOUNT 0
GO
USE DS
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'BROWSE_BY_TITLE' AND type = 'P')
    DROP PROCEDURE BROWSE_BY_TITLE
GO
USE DS
CREATE PROCEDURE BROWSE_BY_TITLE
    (@batch_size_in INT,
    @title_in VARCHAR(50))
AS
    SET ROWCOUNT @batch_size_in
    SELECT * FROM PRODUCTS WHERE TITLE=@title_in
    SET ROWCOUNT 0
GO
USE DS
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'PURCHASE' AND type = 'P')
    DROP PROCEDURE PURCHASE
GO
USE DS
CREATE PROCEDURE PURCHASE
    (@customerid_in INT,
    @number_items INT,
    @netamount_in MONEY,
    @taxamount_in MONEY,
    @totalamount_in MONEY,
    @prod_id_in0 INT = 0,  @qty_in0 INT = 0,
    @prod_id_in1 INT = 0,  @qty_in1 INT = 0,
    @prod_id_in2 INT = 0,  @qty_in2 INT = 0,
    @prod_id_in3 INT = 0,  @qty_in3 INT = 0,
    @prod_id_in4 INT = 0,  @qty_in4 INT = 0,
    @prod_id_in5 INT = 0,  @qty_in5 INT = 0,
    @prod_id_in6 INT = 0,  @qty_in6 INT = 0,
    @prod_id_in7 INT = 0,  @qty_in7 INT = 0,
    @prod_id_in8 INT = 0,  @qty_in8 INT = 0,
    @prod_id_in9 INT = 0,  @qty_in9 INT = 0)
AS
DECLARE
@date_in                  DATETIME,
@neworderid               INT,
@item_id                  INT,
@prod_id                  INT,
@qty                      INT

SET DATEFORMAT ymd
SET @date_in = GETDATE()
--SET @date_in = '2003/10/31'

-- CREATE NEW ENTRY IN ORDERS TABLE
INSERT INTO ORDERS
(  ORDERDATE,
  CUSTOMERID,
  NETAMOUNT,
  TAX,
  TOTALAMOUNT
 )
VALUES
(  @date_in,
  @customerid_in,
  @netamount_in,
  @taxamount_in,
  @totalamount_in
 )

SET @neworderid = @@IDENTITY

-- ADD LINE ITEMS TO ORDERLINES
SET @item_id = 0
WHILE (@item_id < @number_items)
BEGIN
  SELECT @prod_id = CASE @item_id WHEN 0 THEN @prod_id_in0
                       WHEN 1 THEN @prod_id_in1
                       WHEN 2 THEN @prod_id_in2
                       WHEN 3 THEN @prod_id_in3
                       WHEN 4 THEN @prod_id_in4
                       WHEN 5 THEN @prod_id_in5
                       WHEN 6 THEN @prod_id_in6
                       WHEN 7 THEN @prod_id_in7
                       WHEN 8 THEN @prod_id_in8
                       WHEN 9 THEN @prod_id_in9

  SELECT @qty = CASE @item_id WHEN 0 THEN @qty_in0
                  WHEN 1 THEN @qty_in1
                  WHEN 2 THEN @qty_in2
                  WHEN 3 THEN @qty_in3
                  WHEN 4 THEN @qty_in4
                  WHEN 5 THEN @qty_in5
                  WHEN 6 THEN @qty_in6
                  WHEN 7 THEN @qty_in7
                  WHEN 8 THEN @qty_in8
                  WHEN 9 THEN @qty_in9

END

INSERT INTO ORDERLINES
(  ORDERLINEID,
  ORDERID,
  PROD_ID,
  QUANTITY,
  QuANTITY
  -- MORE QUANTITY
  )
VALUES
(  @orderlineid_in,
  @orderid_in,
  @prod_id_in,
  @qty_in,
  @subtotal_in
 )

SET @neworderid = @@IDENTITY

-- ADD LINE ITEMS TO ORDERLINES
SET @item_id = 0
WHILE (@item_id < @number_items)
BEGIN
  SELECT @prod_id = CASE @item_id WHEN 0 THEN @prod_id_in0
                       WHEN 1 THEN @prod_id_in1
                       WHEN 2 THEN @prod_id_in2
                       WHEN 3 THEN @prod_id_in3
                       WHEN 4 THEN @prod_id_in4
                       WHEN 5 THEN @prod_id_in5
                       WHEN 6 THEN @prod_id_in6
                       WHEN 7 THEN @prod_id_in7
                       WHEN 8 THEN @prod_id_in8
                       WHEN 9 THEN @prod_id_in9

  SELECT @qty = CASE @item_id WHEN 0 THEN @qty_in0
                  WHEN 1 THEN @qty_in1
                  WHEN 2 THEN @qty_in2
                  WHEN 3 THEN @qty_in3
                  WHEN 4 THEN @qty_in4
                  WHEN 5 THEN @qty_in5
                  WHEN 6 THEN @qty_in6
                  WHEN 7 THEN @qty_in7
                  WHEN 8 THEN @qty_in8
                  WHEN 9 THEN @qty_in9

END

INSERT INTO ORDERLINES
(  ORDERLINEID,
  ORDERID,
  PROD_ID,
  QUANTITY,
```sql
ORDERDATE
}
VALUES
{
@item_id,
@neworderid,
@prod_id,
@qty,
@dte_in
}
SET @item_id = @item_id + 1
END

SELECT @neworderid
GO

USE DS
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'ROLLUP_BY_CATEGORY_FULL' AND type = 'P')
DROP PROCEDURE ROLLUP_BY_CATEGORY_FULL
GO

USE DS
GO

CREATE PROCEDURE ROLLUP_BY_CATEGORY_FULL
  (@category_in              INT)
AS

  SET DATEFORMAT YMD

  SELECT SUM(OL.QUANTITY * P.PRICE) FROM ORDERLINES OL, PRODUCTS P
  WHERE OL.PROD_ID = P.PROD_ID AND P.CATEGORY = @category_in AND '2003-06-01' <= OL.ORDERDATE AND '2003-07-01' > OL.ORDERDATE;

  SELECT SUM(OL.QUANTITY * P.PRICE) FROM ORDERLINES OL, PRODUCTS P
  WHERE OL.PROD_ID = P.PROD_ID AND P.CATEGORY = @category_in AND '2003-04-01' <= OL.ORDERDATE AND '2003-07-01' > OL.ORDERDATE;

  SELECT SUM(OL.QUANTITY * P.PRICE) FROM ORDERLINES OL, PRODUCTS P
  WHERE OL.PROD_ID = P.PROD_ID AND P.CATEGORY = @category_in AND '2003-07-01' > OL.ORDERDATE;

GO
```