

# ACCELERATING VIRTUALIZED ORACLE 12C PERFORMANCE WITH VSPHERE 5.5 ADVANCED FEATURES FLASH READ CACHE AND VMOTION

## VMware® vSphere® 5.5 with vSphere Flash Read Cache™



IT administrators are always looking for ways to improve and fully utilize their hardware resources. Virtualizing IT infrastructure for critical applications and databases, such as Oracle Database 12c, has become the IT industry trend, providing the ability to condense multiple workloads on a single server. VMware vSphere is a purpose-built hypervisor designed to provide the performance, reliability, and flexibility that these mission-critical applications require. With new features such as vSphere Flash Read Cache™ (vFRC) in vSphere 5.5, VMware can improve Oracle Database 12c performance while maintaining the reliability features you have come to expect from the platform, including VMware vMotion.

In the Principled Technologies labs, we set up a four-node VMware vSphere 5.5 cluster using Cisco UCS B200 M3 blade servers and EMC VMAX 10K storage. We ran 10 virtual machines (VMs) on the cluster, each with its own Oracle Database 12c application, and ran a mix of database workloads simultaneously to gather baseline performance data. Then, we enabled the new vFRC feature specifically on the OLAP workloads and ran the tests again. We found that vSphere 5.5 with vSphere Flash Read Cache-enabled VMs decreased the time it took to run an OLAP workload by up to 14 percent. Additionally, we demonstrated the tried-and-true VMware vMotion functionality when we enabled the vFRC feature and moved VMs from one server to



another. The vFRC-enabled VMs transitioned seamlessly to the other hosts while continuing to cache on the destination host.

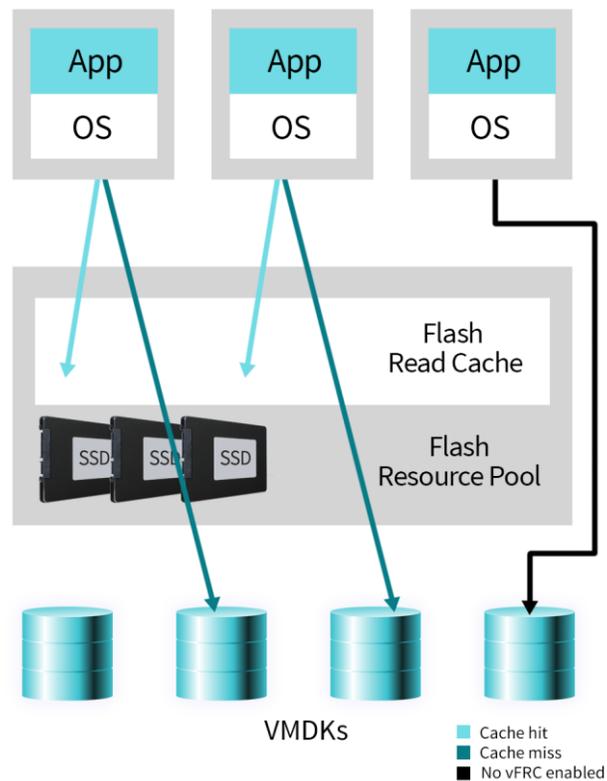
## VSPHERE FLASH READ CACHE BOOSTED ORACLE PERFORMANCE

### Benefits on the vFRC-enabled VM

Maximizing the performance of your virtualized critical Oracle Database 12c applications is crucial to the success of your business. As we found in our tests, vSphere Flash Read Cache in VMware vSphere 5.5 can improve performance without sacrificing the tools that have become critical to your infrastructure management, such as VMware vMotion. VMware vFRC is designed to lower application latency by virtualizing server-side flash storage to provide a high-performing read cache layer.<sup>1</sup> As data on a vFRC-enabled virtual machine disk (VMDK) is read, vFRC copies the data to a flash resource pool that is comprised of one or more high-performance, enterprise flash devices at the individual ESXi host level. As repeated reads begin to occur over time on the vFRC-enabled VMDK, the VM accesses the data from the flash resource pool, bypassing the VMDK and the underlying physical storage. Not only can this offload of data from the storage onto the local server benefit vFRC enabled VMs, it can also relieve shared storage resources in a mixed workload environment, indirectly improving storage access performance across the board. See Figure 1 for a detailed diagram of how data is read by vFRC-enabled VMs in vSphere 5.5.

---

<sup>1</sup> To learn more about vSphere Flash Read Cache, visit [www.vmware.com/products/vsphere/features-flash](http://www.vmware.com/products/vsphere/features-flash).

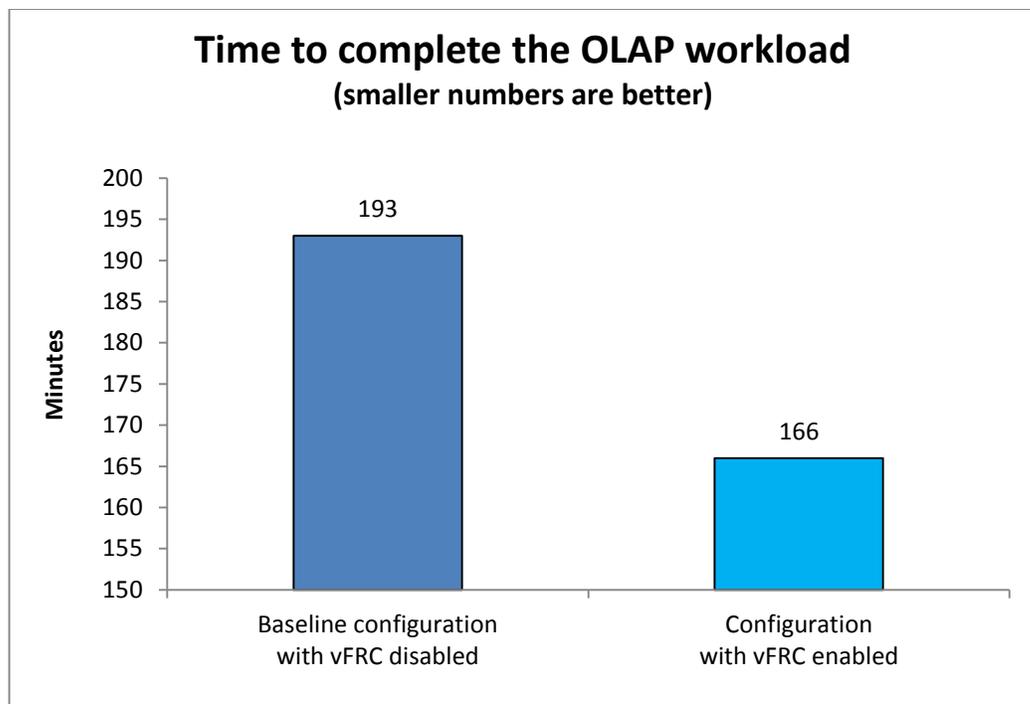


**Figure 1: Detailed view of how vSphere Flash Read Cache works in vSphere 5.5.**

We put the performance of vFRC to the test on a VMware cluster comprised of Cisco UCS B200 M3 servers with EMC VMAX 10K storage. This combination of VMware software, Cisco servers, and EMC storage delivered promising results for our virtualized Oracle Database 12c workloads. See [Appendix A](#) for system configuration details, and [Appendix B](#) and [Appendix C](#) for detailed testing steps. See the section “[vSphere Flash Read Cache configuration](#)” in Appendix B for details on our vFRC configuration.

As Figure 2 shows, the baseline configuration with vFRC *disabled* did not perform as well as the configuration with vFRC *enabled*. The baseline configuration took 3 hours and 13 minutes (or 193 minutes) to complete the OLAP test while the configuration with vFRC-enabled VMs took 2 hours and 46 minutes (or 166 minutes). Enabling vFRC decreased the time it took to complete the TPC-H-like OLAP workload by 14 percent.

**Figure 2: Enabling vSphere Flash Read Cache decreased the time it took to complete the TPC-H-like OLAP workload by 14 percent.**



## New feature, same performance during vMotion

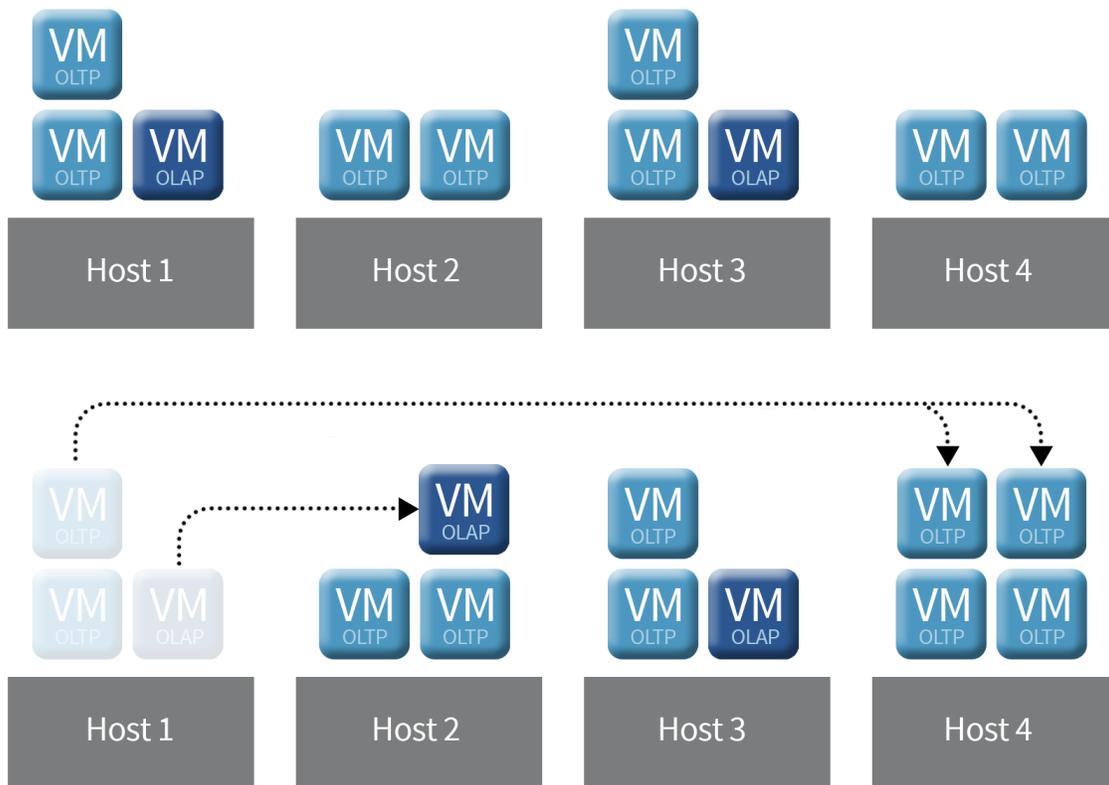
### In our labs...

We achieved max throughput of up to 23 Gb/s during vMotion.

Additionally, the vFRC-enabled OLAP VM migration took 2 minutes and 6 seconds to complete.

Servers need occasional maintenance, and the ability to live migrate important virtualized database workloads is key in avoiding application downtime. This is where the flexibility of virtualization with VMware vSphere can benefit businesses. VMware vMotion allows you to perform these live migrations of VMs from one server in a cluster to another, without causing your workload performance to take a hit.

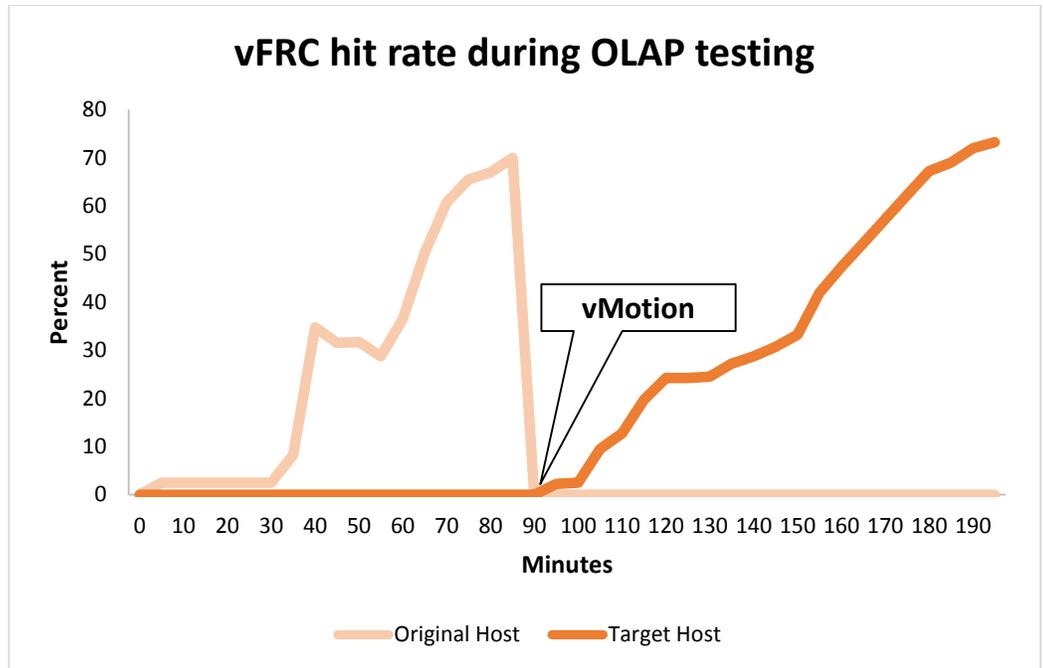
You can continue to utilize this familiar functionality of vMotion with vSphere Flash Read Cache, as it checks destination-host compatibility for cache devices when choosing to migrate vFRC-enabled VMs. Scheduling migrations via vMotion in vSphere 5.5 is the same as previous versions, with the additional choice to migrate the cache contents with the VM or allow the VM to re-cache once the migration is complete. We used vMotion to migrate the VMs off one host containing one vFRC-enabled OLAP VM and two OLTP VMs that were not vFRC enabled. While there are benefits to migrating the cache with the VM, we chose to allow the cache to re-warm on the destination server, illustrating that even in situations where the cache has to re-warm, workloads still benefit from vFRC. The vFRC configuration migrated and continued caching on the destination server without issue. Figure 3 shows how we performed vMotions in our testing.



**Figure 3: Moving VMs from one host to another with vMotion.**

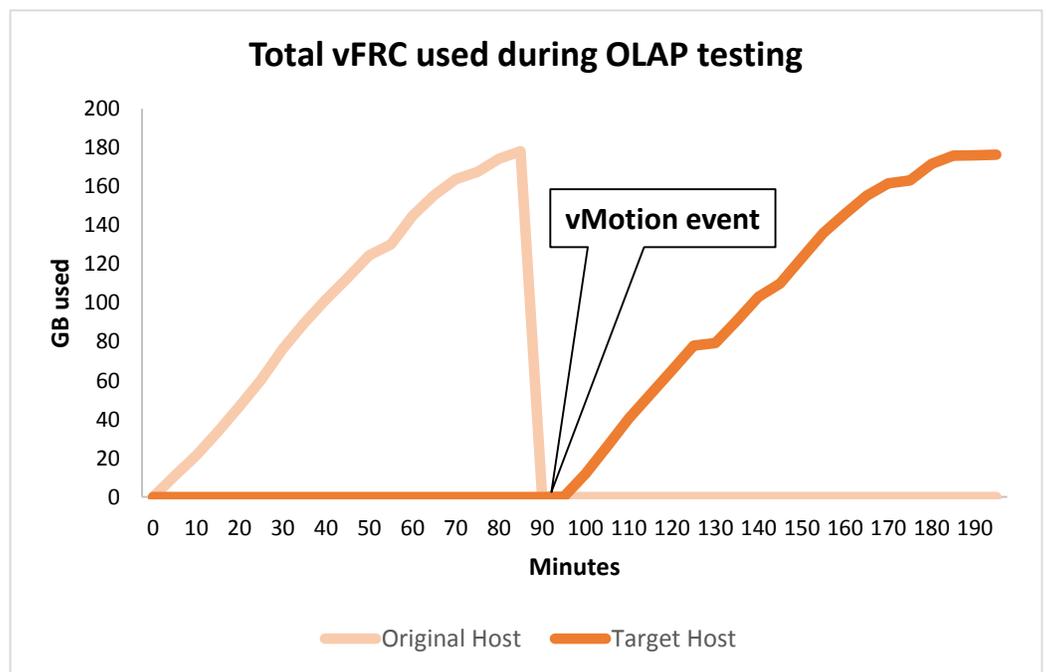
We measured the vFRC hit rate percent on our original host before the vMotion event and on the target host after the vMotion event. Hit rate is a good measurement of how well vFRC is performing. We looked at the hit rate percentage in five-minute intervals during testing. Figure 4 shows the vFRC hit rate percent during our TPC-H-like OLAP testing. Our vMotion event took 2 minutes and six seconds to complete and started at the 90-minute mark.

Figure 4: The vFRC hit rate percentage for the original host before the vMotion event and for the target host after the vMotion event.



During the same test, we also measured the total GB of vFRC used by the VMs. Although similar to the hit rate results, the number of GB used by the VMs presents raw, quantifiable data that would be subject to scaling in your datacenter. We looked at the total GB of vFRC in five-minute intervals during testing and at the same times when we monitored the hit rate. Figure 5 shows the amount of GB of vFRC used by the VMs during our TPC-H-like OLAP testing. Length and start time for our vMotion event are the same as Figure 3.

Figure 5: The total GB of vFRC used by the VMs for the original host before the vMotion event and for the target host after the vMotion event.



## WHAT THIS MEANS FOR YOU

Administrators can view the benefits of vFRC-enabled VMs as they apply to their specialties or focus for their infrastructure. In the following sections, we illustrate the impact of our results tailored to those specialties or focuses.

### What this means for VMware vSphere admins



To keep current with new technology and improvements, VMware looks to bring increased functionality to vSphere with every release. A major change in vSphere 5.5 comes in the addition of vFRC. With VMware vSphere 5.5, you now have access to more performance-enhancing features, like vFRC, while using the same interface and management tools you already use in your virtualized environment. You can upgrade your infrastructure to version 5.5 from 5.1 and start realizing the benefits of these new features immediately. VMware vSphere continues to be the flagship for VMware and it is still a preferred choice of virtualization platforms in various-sized datacenters.

The new vSphere Flash Read Cache feature we tested enables pooling of flash-based devices into a single vSphere Flash Resource to speed up performance of read-intensive workloads. As our tests results show, VMware vSphere 5.5 can provide a powerful platform for critical Oracle Database 12c applications and improve performance with its new features, while still being able to depend on the reliability and speed of vMotion.

### What this means for Oracle Database 12c admins



Administrators dealing with Oracle Database applications have two main concerns: performance and reliability. The new vSphere Flash Read Cache feature helps address the performance concern. For read-heavy OLAP workloads, vSphere 5.5 with vFRC enabled can increase the performance of your decision support systems. Any performance increase can translate to getting more from your hardware, which can mean a delay in upgrading current hardware/purchasing new hardware and any associated costs, such as Oracle licensing.

Business-critical databases, whether virtualized or bare metal, cannot go down or work grinds to a halt. With reliability features such as VMware vMotion, which lets you move VMs from one server to another for maintenance events, and performance-enhancing features like vFRC, Oracle Database applications can keep working and improving even if maintenance is needed on your hardware.

### What this means for Cisco server admins



Enterprise-class servers, such as the Cisco UCS B200 M3 blade servers we used in our study, can help deliver high levels of performance and density for virtualized Oracle Database 12c workloads running on VMware vSphere 5.5 and vFRC. The Cisco UCS chassis we used in testing is capable of providing up to 80 Gbps of bandwidth

without a switch to any blade where vMotion can occur. The VIC 1240 adapter that we used for the UCS infrastructure enables 40 Gbps to each blade by default. With only a single FCoE connection to the Fabric Interconnects, the VIC 1240 can burst to 10 Gbps per Fabric Interconnect or 20 Gbps per blade. This allows you the flexibility you need when using vFRC or performing vMotion. With our configuration, we were able to push vMotion throughput up to 23 Gbps. To see how we setup and used our Cisco components, see the section “[Hardware and Software](#)” in Appendix B.

The Cisco UCS infrastructure features a converged fabric where all systems management and configuration originates from a pair of redundant Fabric Interconnects (FI) to allow management of large-scale deployments and migrations from a single location, easing the job of server admins.

For our flash device, we used an LSI 400GB SLC WarpDrive mezzanine card, sold by Cisco for their UCS blade servers. By adding the LSI WarpDrive to our flash resource pool, we not only gained added capacity for read cache, but we also gained the reliability and durability of SLC solid-state technology, ensuring great underlying hardware performance for vFRC.

By pairing this Cisco architecture with the performance enhancing and reliability features of VMware vSphere 5.5 and vFRC, you can ensure you get the most out of mission-critical workloads.

## What this means for EMC VMAX storage admins



The storage solutions that storage administrators choose can greatly affect the performance of critical Oracle Database 12c workloads running on vFRC-enabled VMs. The EMC VMAX 10K we used in our tests is an enterprise-class storage array that is tiered with EFD, FC, and SATA disks leveraging FAST technology. This ability to choose from various tiered I/O performance levels ensures you get the storage I/O needed to run a virtualized mixed Oracle Database 12c environment, regardless of the specific storage demands for each virtual workload. In a virtualized environment like a typical VMware vSphere 5.5 cluster, the ability to provide different levels of performance capabilities to each workload while still carefully managing storage resources is crucial. VMAX 10K features like QOS service levels, dynamic host I/O limits, and storage tiering make it easy to ensure hypervisor hosts and virtual machines that need more I/O get the storage resources they need, while still providing reliable service to other hosts in the environment.

With the read caching capabilities of vSphere Flash Read Cache, frequently read data is moved from external storage to a local flash device on a server. The vFRC enabled workloads and increased reads being serviced by the underlying flash device have a closer data locality as more storage I/O becomes available. This additional headroom allows storage administrators to see potential increases in storage

performance for other workloads in the environment during times of peak utilization. To see how we configured our storage layout for the VMAX 10K, see the section titled [“Storage Layout”](#) in Appendix B.

## IN CONCLUSION

Strong Oracle Database 12c performance is vital to the state of your business. Virtualizing such important workloads requires a reliable and high-performing virtualization platform, along with the right servers and storage. EMC, Cisco and VMware offer proven technologies to meet this need. In addition, newer technologies like vFRC can have a positive impact on database performance by offloading some of the storage I/O onto the local server. This can be beneficial to the intended application and has the potential to improve all applications in a mixed workload environment over time by relieving pressure on shared storage resources.

In our tests, we found that the new release of VMware vSphere 5.5 provided a new feature, vSphere Flash Read Cache, that decrease TPC-H-like OLAP workload processing time by 14 percent. We also found that running these workloads on Oracle Database 12c with the new feature didn't affect the ability of administrators to complete routine vMotion tasks; with vSphere Flash Read Cache enabled during a vMotion, the migration went smoothly and vFRC continued to cache after the migration completed. This means that the combination of VMware vSphere 5.5 platform, Cisco UCS B200 M3 servers, and EMC VMAX 10K storage was able to provide improved Oracle Database 12c performance using the new vSphere Flash Read Cache feature, which improves the reliability and database response times you deliver for customers and employees alike.

## APPENDIX A – SYSTEM CONFIGURATION INFORMATION

Figure 6 provides detailed configuration information for the test systems.

System	4x Cisco UCS B200 M3 server
<b>General</b>	
Number of processor packages	2
Number of cores per processor	8
Number of hardware threads per core	2
System power management policy	Default
<b>CPU</b>	
Vendor	Intel®
Name	Xeon®
Model number	E5-2680
Stepping	7
Socket type	LGA2011
Core frequency (GHz)	2.7
Bus frequency	8.00 GT/s
L1 cache	32KB +32KB
L2 cache	256KB per core
L3 cache	20MB
<b>Platform</b>	
Vendor and model number	Cisco UCS B200 M3
Motherboard model number	Cisco FCH1607GV4
BIOS name and version	Cisco B200M3.2.1.1a.0.121720121447
BIOS settings	Default
<b>Memory modules</b>	
Total RAM in system (GB)	320
Vendor and model number	16x Cisco UCS-MR-1X162RY-A16, 8x Cisco UCS-MR-1X082RY-A
Type	PC3-12800
Speed (MHz)	1,600
Speed running in the system (MHz)	1,333
Size (GB)	(16x) 16, (8x) 8
Number of RAM module(s)	24 (16 + 8)
Chip organization	Double-sided
Rank	Dual
<b>Hypervisor</b>	
Name	VMware vSphere 5.5.0
Build number	1331820
Language	English
<b>RAID controller</b>	
Vendor and model number	LSI® MegaRAID SAS 2004
Firmware version	20.10.1-0100

System	4x Cisco UCS B200 M3 server
<b>Hard drives</b>	
Vendor and model number	Seagate A03-D146GC2
Number of drives	2
Size (GB)	146
RPM	15,000
Type	SAS
<b>SSD cache drive</b>	
Vendor and model number	LSI UCSB-F-LSI-400S SLC WarpDrive®
Number of drives	1
Size (GB)	400
RPM	n/a
Type	PCI-E
<b>Converged I/O adapters</b>	
Vendor and model number	Cisco UCSB-MLOM-40G-01, Cisco UCS-VIC-M82-8P
Type	mLOM, Mezzanine
<b>Virtual machine operating system</b>	
Name	Oracle Enterprise Linux Release 6.5
Kernel	3.8.13-26.2.1.el6uek.x86_64
Language	English
Database software	Oracle Database 12c Build 12.1.0.1.0
<b>Database benchmarks</b>	
Benchmark 1	HammerDB v2.15
Benchmark 2	DVD Store 2.1

Figure 6: Configuration information for the systems used in our tests.

Figure 7 provides the firmware information for the Cisco hardware we used in our tests.

UCS 5108 chassis	Firmware version
UCS Manager	2.2(1b)
UCS 2208XP IO Module 1 & 2	2.2(1b)
UCS 6248UP Fabric Interconnect 1 & 2	5.2(3)N2(2.21b)
UCS B200 M3 blades	Firmware version
BIOS	B200M3.2.2.1a.0.111220131105
CIMC Controller	2.2(1b)
UCS VIC 1240	2.2(1b)

Figure 7: Firmware information for the Cisco hardware used in our tests.

## APPENDIX B – WHAT WE TESTED

### Hardware and software

In our test bed, we configured the Cisco UCS 5108 Blade Server chassis with four cables coming from each FEX (eight in total), going into two UCS 6248UP Fabric Interconnects (FIs). We then cabled each FI via four 10Gb Ethernet port and one 8Gb FC port to two Cisco Nexus™ 5548UP switches, with two Ethernet links from each FI connected to each switch, resulting in a fully redundant infrastructure. We aggregated each set of four Ethernet ports into a port channel to ensure maximum bandwidth. Figure 8 illustrates our test bed.

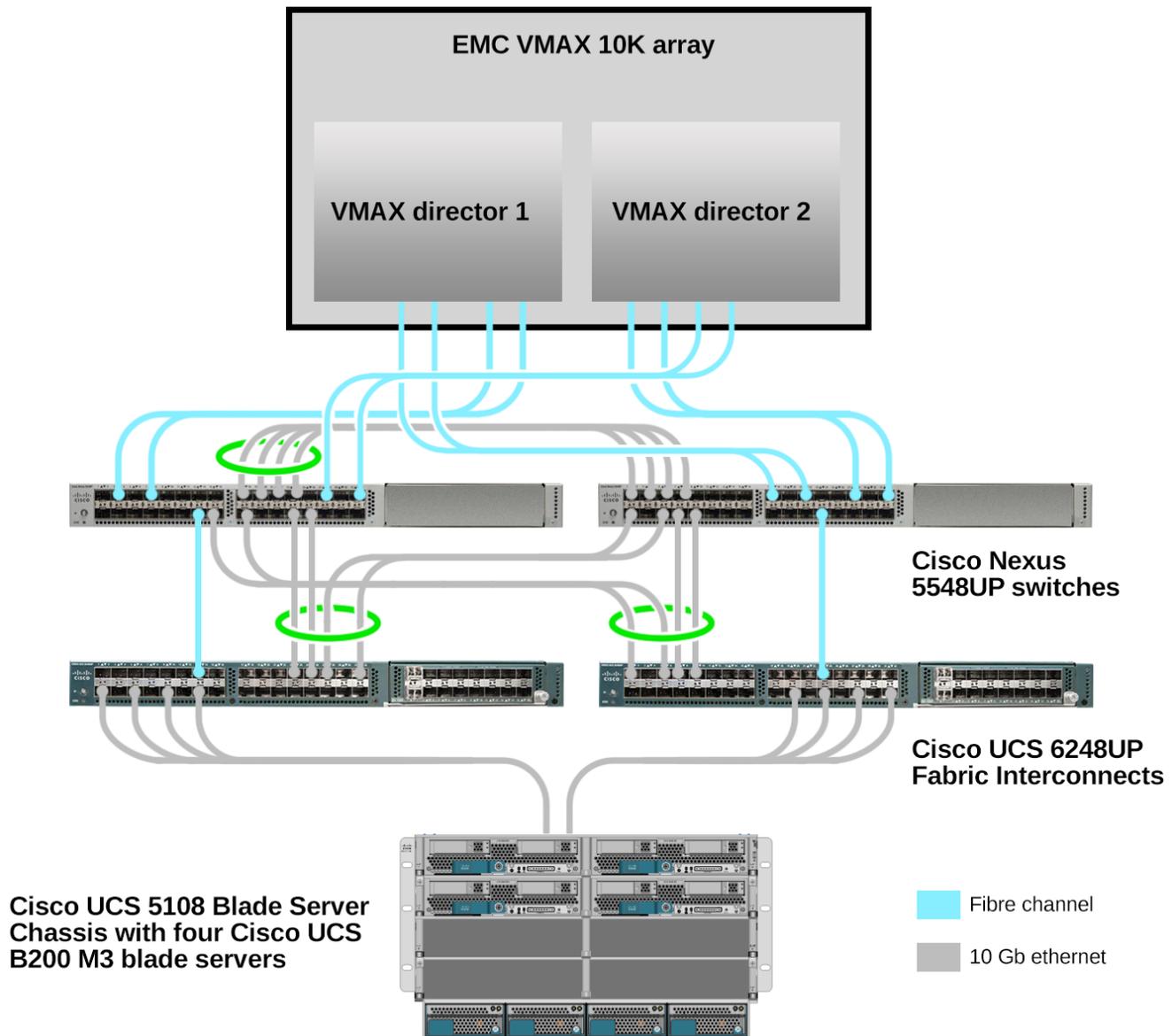


Figure 8: The test bed used in testing and how components were connected.

We then configured four Cisco UCS B200 M3 blade servers with VMware vSphere 5.5. On a separate rack server, we configured a VMware vCenter server, connected it to the UCS chassis network, and created a cluster in the VMware

vSphere Web Console using the four blade servers. We created two vSphere vSwitches on each blade server: the first for VM management and Oracle Application connections and the second for vMotion traffic. Each vMotion vSwitch had a maximum MTU of 9,000 bytes and used four uplink ports (physical NICs). The management and Oracle Application vSwitches used two of those uplink ports.

For the vMotion network specifically, we setup multi-NIC vMotion on the vSphere hosts according to VMware best practices.<sup>2</sup> In the Networking inventory area of the vSphere client, we created two VMkernel on the vMotion vSwitch. We assigned two ports on the vMotion subnet to each, and configured teaming and failover. For the first VMkernel, we assigned the first uplink as active and the second as standby. On the second, we assigned the first uplink as standby and the second as active. The vMotion network used Jumbo Frames (MTU 9000).

## Virtual machine

We had ten VMs total: two OLAP VMs to run the TPC-H-like workload and eight OLTP VMs to run the TPC-C-like workload. The two OLAP VMs had eight vCPUs and 200GB of RAM each. The eight OLTP VMs each had two vCPUs and 48GB of RAM. The OLAP VMs had one OS VMDK, four data VMDKs, four temp VMDKs, and four redo log VMDKs. We split the four VMDKs in each group across four LUNs on our storage. The OLTP VMs each had one OS VMDK, one data VMDK, and one redo log VMDK. Our VM virtual NICs used the VMXNET3 type, as it offers the latest paravirtualized benefits for VM NICs.

## vSphere Flash Read Cache configuration

For our vSphere Flash Read Cache configuration, we configured a flash resource pool on each host, comprised of a single LSI 400GB SLC WarpDrive installed in each Cisco UCS B200 M3 server. At the VM level, we configured the four data VMDKs on each OLAP workload VM to use vFRC. We divided the available capacity in the flash resource pool by four (92 GB) and configured each vFRC-enabled data VMDK to use that much cache capacity. For our cache block size, we set each VMDK cache to a 32KB block size to match the configured Oracle database block size of our OLAP workload. We did not configure vFRC for our OLTP workload VMs. See Figure 9 for a detailed look at our OLAP vFRC configuration.

---

<sup>2</sup> [kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=2007467](https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2007467)

# OLAP vFRC configuration

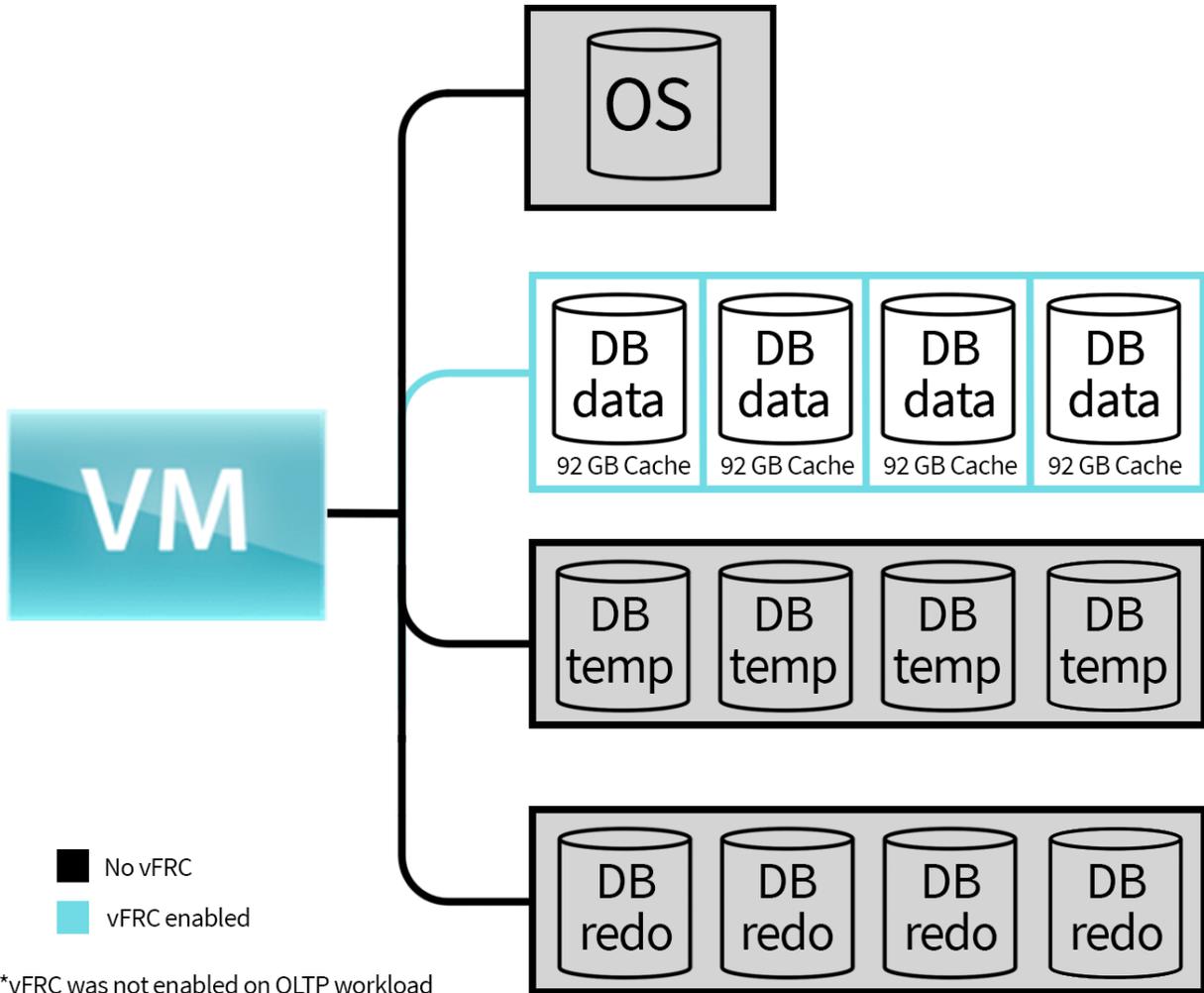


Figure 9: The vFRC configuration for the OLAP VMs used in our testing.

## Test tools

HammerDB is an open-source benchmark tool that tests the database performance of many leading databases, including Oracle Database, Microsoft® SQL Server®, PostgreSQL, MySQL™, and more. The benchmark includes two built-in workloads derived from industry-standard benchmarks: a transactional (TPC-C-like) workload and a data warehouse (TPC-H-like) workload. For this study, we used the data warehouse workload. Our tests were not official TPC results and are not comparable in any manner. For more information about HammerDB, visit [hammerora.sourceforge.net](http://hammerora.sourceforge.net).

We used the DVD Store Version 2.1 (DS2) benchmarking tool for our TPC-C like workload. DS2 models an online DVD store, where customers log in, search for movies, and make purchases. DS2 reports these actions in orders per minute (OPM) that the system could handle, to show what kind of performance you could expect for your customers.

The DS2 workload also performs other actions, such as adding new customers. For more information about the DS2 tool, see [www.delltechcenter.com/page/DVD+Store](http://www.delltechcenter.com/page/DVD+Store).

## Storage layout

### Physical and virtual storage

In our labs, we configured the EMC VMAX 10K array according to best practices. We coordinated with EMC VMAX 10K engineers on management access, cabling, tiering, and monitoring. Figure 10 shows how we connected the components.

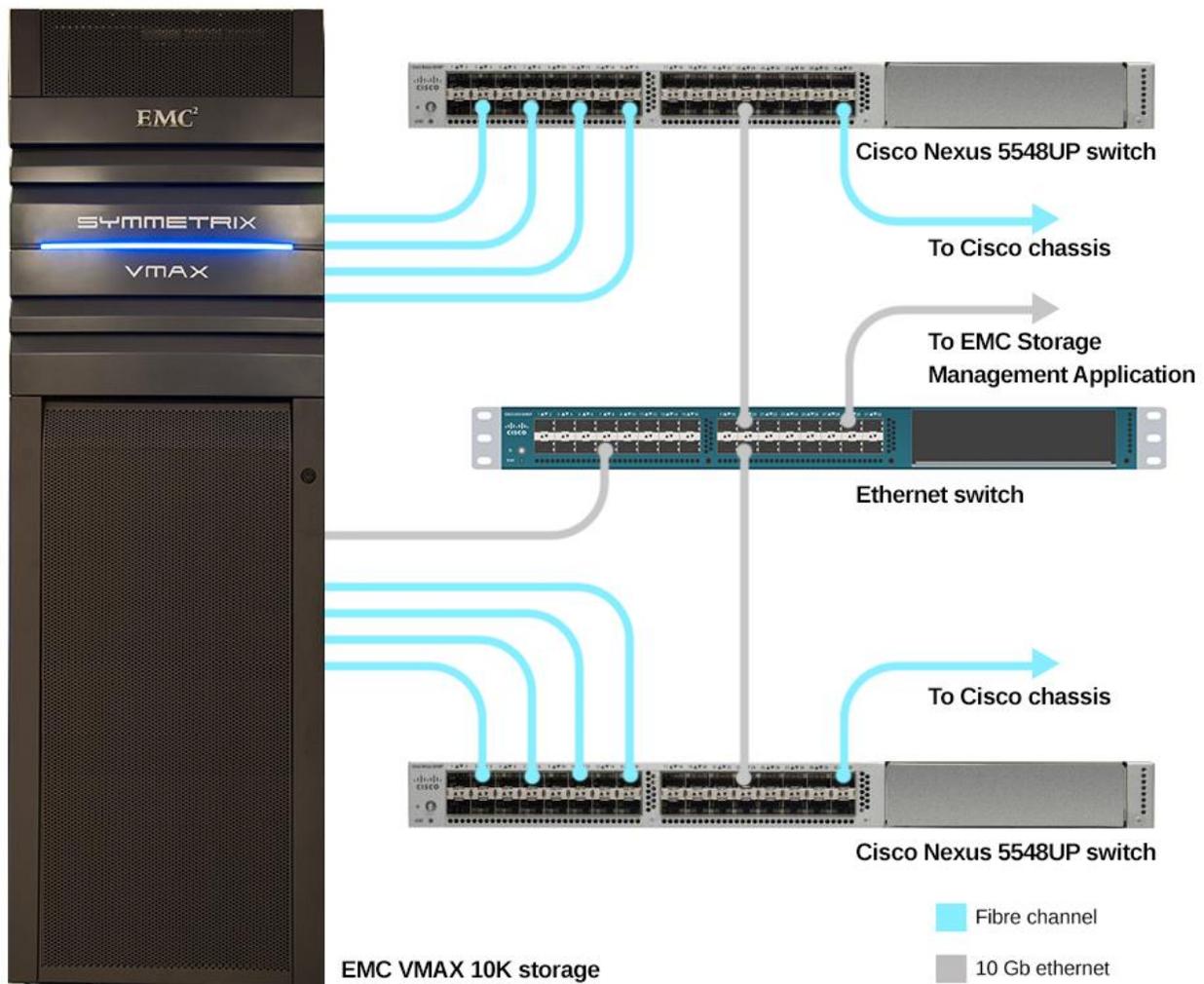


Figure 10: Diagram of the VMAX 10K

We followed EMC best practices for allocating high-performance storage volumes for the vSphere VMDKs associated with the Oracle data, temporary tablespaces, and redo logs. Though we had three configured tiers on our VMAX 10K, we used two tiers of storage in our tests: a top performance tier comprised of EFD and FC disks, and a middle performance tier of EFD, FC, and SATA disks. From the VMAX 10K management console, we created four 512GB volumes

in the top tier and four 4TB volumes in the middle tier. The volumes in the top tier held the virtual disks containing the Oracle OLTP and OLAP databases as well as the OLTP redo logs. The middle tier volumes held virtual disks containing operating system data and the remaining Oracle disks (system files, OLAP temp tablespace, and OLAP redo logs). See Figure 11 for the storage layout of OLAP and OLTP VMs.

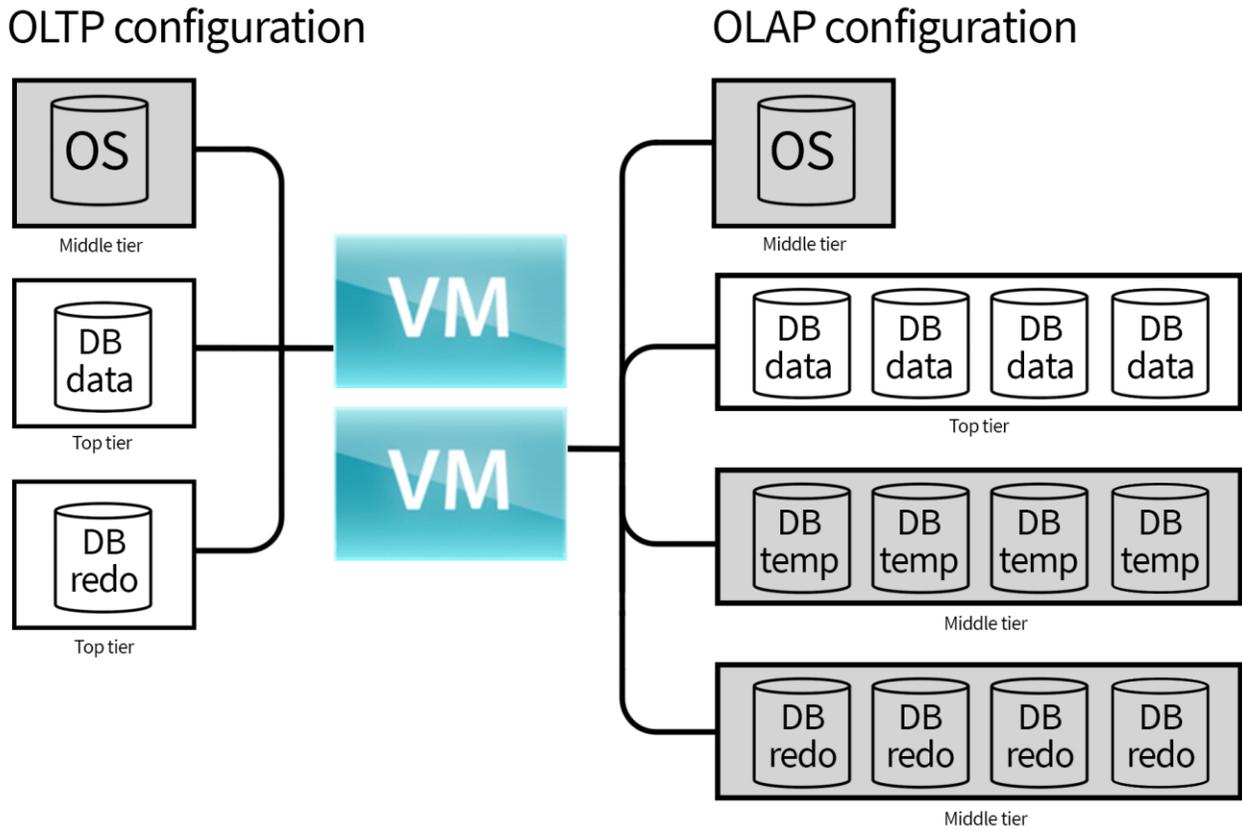


Figure 11: Logical layout of Oracle storage on the EMC VMAX 10K.

## Performance test scenarios and workflow

### Baseline

To determine the impact vFRC and vMotion have on performance, we needed to develop a baseline score. To do so, we ran our TPC-H-like and TPC-C-like workloads against our ten VMs without any other factor involved. We gathered esxtop data, the benchmark scores, and the time to run the TPC-H-like workload to establish our baseline results.

### vFRC enabled

Once we had our baseline scores, we wanted to determine the impact of vFRC on performance scores. We reset everything to default states and enabled vFRC. We split the full amount of the available cache equally across the four data VMDKs on the two OLAP VMs. We then ran our workloads against the ten VMs as we did in the baseline run. We gathered esxtop data, the benchmark scores, the time to run the TPC-H-like workload, and the cache data to determine our new results.

### vFRC and vMotion event

Lastly, we wanted to gauge the effect of a vMotion event on the performance with vFRC enabled, and prove that vFRC and vMotion can work together. After resetting all of our testbed to the default state, we again setup and ran the same vFRC test as our previous run. Using the time to complete the TPC-H-like workload in our previous run, we marked the halfway point in our run, and performed a vMotion event on one OLAP and two OLTP VMs. We then let the workloads finish on their new hosts. We gathered esxtop data, benchmark scores, time to run the TPC-H-like OLAP workload, cache data from both the destination and the target hosts, and vMotion network data to determine our new results.

## APPENDIX C – HOW WE TESTED

### Installing VMware vSphere 5.5 (ESXi) on the Cisco UCS B200 M3 blades

1. Insert the disk, and boot from disk.
2. On the Welcome screen, press Enter.
3. On the End User License Agreement (EULA) screen, press F11.
4. On the Select a Disk to Install or Upgrade Screen, select the relevant volume on which to install ESXi, and press Enter.
5. On the Please Select a Keyboard Layout screen, press Enter.
6. On the Enter a Root Password Screen, assign a root password and confirm it by entering it again. Press Enter to continue.
7. On the Confirm Install Screen, press F11 to install.
8. On the Installation complete screen, press Enter to reboot.
9. Repeat steps 1-8 for each B200 M3 blade.

#### Configuring ESXi after Installation

1. On the ESXi 5.5 screen, press F2, enter the root password, and press Enter.
2. On the System Customization screen, select Troubleshooting Options, and press Enter.
3. On the Troubleshooting Mode Options screen, select enable ESXi Shell, and press Enter.
4. Select Enable SSH, press Enter, and press ESC.
5. On the System Customization screen, select Configure Management Network.
6. On the Configure Management Network screen, select IP Configuration.
7. On the IP Configuration screen, select set static IP, enter an IP address, subnet mask, and default gateway, and press Enter.
8. On the Configure Management Network screen, press Esc. When asked to apply the changes, type Y.
9. Repeat steps 1-8 for each B200 M3 blade.

For our vCenter management server, we deployed the vCenter Appliance on a separate ESXi host and configured it for our environment.

#### Configuring VM networking on ESXi

1. Log into the vSphere Web Client with the administrator credentials and navigate to the B200 M3 host.
2. In the Manage tab, click on Networking, and in the Virtual switches pane, click the Add host networking button.
3. Select Physical Network Adapter, and click Next.
4. Select New standard switch, and assign to it two of the B200 M3's ports connected to the physical test network.
5. Click Next.
6. Provide a label for the switch, and click Next.
7. Click Next.
8. Click Finish.
9. Select VMkernel adapters in the pane on the left.
10. Click Add Networking.
11. Choose Virtual machine Port Group for a Standard Switch, and click Next.
12. Select the switch you just created, and click Next.
13. Label the network and assign the appropriate VLAN ID, and click Next.

14. Click Finish.
15. Click Add Networking again, and choose VMkernel Network Adapter.
16. Choose the switch you created earlier, and click Next.
17. Label the network, choose the appropriate VLAN ID, check Management traffic, and click Next.
18. Use static IPv4 settings, and set the appropriate address. Click next.
19. Click Finish.
20. Repeat steps 1 – 8 to create a second vSwitch assigning 4 physical ports instead of 2.
21. Click on the new vSwitch, and click Edit.
22. Change the MTU to 9000 Bytes, and click OK.
23. Repeat steps 15 – 19 to create four vMotion VMkernels (check vMotion traffic instead of Management).
24. For each vMotion VMkernel, click edit and change the NIC settings to 9000 MTUs.
25. Click on Virtual switches in the left hand pane.
26. Click on the vMotion vSwitch, and highlight the first vMotion VMkernel.
27. Click Edit.
28. Go to Teaming and failover, and check Override.
29. Choose one vmnic to be active, and set the rest as Standby.
30. Click OK.
31. Repeat steps 27 through 30 for each of the other three VMkernels each time choosing a different vmnic to remain active, giving each VMkernel its own vmnic.

### Creating the first VM

1. From the B200 M3 host in the vSphere Web Client, navigate to the Virtual Machines page.
2. Click the New Virtual Machine button.
3. At the Select a creation type screen, select Create a new virtual machine, and click Next.
4. Assign a name to the virtual machine, and click Next.
5. Select the B200 M3 host to run the virtual machine, and click Next.
6. Select the first assigned OS Datastore on the external storage, and click Next.
7. At the Select compatibility screen, select ESXi 5.5 and later, and click Next.
8. Choose Linux, and choose Oracle Linux 4/5/6 (64-bit), and click Next.
9. If this is an OLAP VM, choose four CPUs; for an OLTP VM, choose two CPUs.
10. If the is an OLAP VM, provide 200GB RAM; for an OLTP VM, provide 48GB RAM.
11. For New Network, select the switch made previously, and choose the VMXNET3 adapter type.
12. For the first hard disk, assign 60GB for the OLAP VMs; for an OLTP VM, assign 40GB , thick provision eager zeroed, and the virtual device node SCSI(0:0).
13. From the New device: drop-down menu, select SCSI Controller.
14. Change the new SCSI controller's type to VMware Paravirtual.
15. Add VHDs to the OLTP VMs
  - a. Add a new hard disk, assign 120 GB, thick provision eager zeroed, and the virtual device node SCSI(1:0). Also change its location to an appropriate datastore. This will be the Oracle data volume.
  - b. Add a new hard disk, assign 10 GB, thick provision eager zeroed, and the virtual device node SCSI(1:1). Also change its location to an appropriate datastore. This will be the Oracle redo logs volume.
16. Add VHDs to the OLAP VMs

- a. Add four new 120GB VHDs, Thick provision lazy zeroed, assigned to an appropriate datastore. These will hold the Oracle data.
  - b. Add four new 10GB VHDs, Thick provision lazy zeroed, assigned to an appropriate datastore. These will hold the Oracle redo logs.
  - c. Add four new 30GB VHDs, Thick provision lazy zeroed, assigned to an appropriate datastore. These will hold the Oracle temp tablespace.
17. Click Next
  18. Click Finish.
  19. Start the VM.
  20. Attach the Oracle Enterprise Linux 6.4 ISO image to the VM and install Oracle Enterprise Linux 6.4 on your VM.

## Oracle Linux 6.5 and Oracle Database 12c

We configured each VM with Oracle Linux 6.5 and Oracle Database 12c. For our OLTP VMs, we used a basic install of Oracle Database 12c and used the local file system for database files: one disk for data; a second disk for redo logs. For our OLAP VMs, we first installed Oracle Grid Infrastructure and utilized Oracle Automatic Storage Management to create three four-disk ASM groups for data, redo logs, and temp table spaces. We then installed and configured Oracle Database 12c utilizing these ASM disk groups for our database files.

### Installing Oracle Linux 6.5

1. Insert the Oracle Linux 6.5 DVD into the server, and boot to it.
2. Select Install or upgrade an existing system.
3. If you are unsure of the fidelity of the installation disk, select OK to test the installation media; otherwise, select Skip.
4. In the opening splash screen, select Next.
5. Choose the language you wish to use, and click Next.
6. Select the keyboard layout, and click Next.
7. Select Basic Storage Devices, and click Next.
8. Select Fresh Installation, and click Next.
9. Insert the hostname, and select Configure Network.
10. In the Network Connections menu, configure network connections.
10. After configuring the network connections, click Close.
11. Click Next.
12. Select the nearest city in your time zone, and click Next.
13. Enter the root password, and click Next.
14. Select Use All Space, and click Next.
15. When the installation prompts you to confirm that you are writing changes to the disk, select Write changes to disk.
16. Select Software Basic Server, and click Next. Oracle Linux installation begins.
17. When the installation completes, select Reboot to restart the server.

### Installing VMware Tools

1. Install guest tools or agents.
2. (VMware only) Install VMware Tools on the guest:
  - Right-click the VM in the Web Client, and select Install/Upgrade VMware Tools.
  - Log onto the guest as root

- Mount the CDROM device:  
# mount -o ro /dev/cdrom /mnt
- Untar VMware Tools into a temporary directory:  
# tar -C /tmp -zxf /mnt/VMwareTools-9.4.0-1280544.tar.gz
- Run the install script and accept the defaults:  
# /tmp/vmware-tools-distrib/vmware-install.pl
- Follow the prompts to configure and install VMware tools.
- The installer will automatically load the NIC drivers, create a new initrd, and unmount the CD.
- Reboot the VM.

### Initial configuration tasks

Complete the following steps to provide the functionality that Oracle Database requires. We performed all of these tasks as root.

1. Disable firewall services. In the command line (as root), type:

```
# service iptables stop
# chkconfig iptables off
# service ip6tables stop
# chkconfig ip6tables off
```

2. Set SELinux:

```
# vi /etc/selinux/config
SELINUX=permissive
```

3. Modify /etc/hosts to include the IP address of the internal IP and the hostname.

4. Edit 90-nproc.conf:

```
# vim /etc/security/limits.d/90-nproc.conf
```

Change this:

```
* soft nproc 1024
```

To this:

```
* - nproc 16384
```

5. (OLAP VM only) Enable huge pages by adding these lines to /etc/sysctl.conf:

```
vm.nr_hugepages=61440
vm.hugetlb_shm_group=54321
```

6. Install 12c RPM packages, resolve package dependencies and modify kernel parameters:

```
# yum install oracle-rdbms-server-12cR1-preinstall -y
```

7. Install automatic system tuning for database storage through yum:

```
# yum install tuned
# chkconfig tuned on
# tuned-adm profile enterprise-storage
```

8. Using yum, install the following prerequisite packages for Oracle Database:

```
# yum install elfutils-libelf-devel
# yum install xhost
# yum install unixODBC
# yum install unixODBC-devel
# yum install oracleasm-support oracleasm lib oracleasm
```

9. Create the oracle user account and groups and password:

```
# groupadd -g 1003 oper
# groupadd -g 1004 asmadmin
# groupadd -g 1005 asmdba
# groupadd -g 1006 asmoper
# usermod -G dba,oper,asmadmin,asmdba,asmoper oracle
# passwd oracle
```

#### 10. Create the /u01 directory for Oracle inventory and software and give it to the oracle user:

```
# mkdir -p /u01/app/oracle/product/12.1.0/grid
# mkdir -p /u01/app/oracle/product/12.1.0/dbhome_1
# chown -R oracle:oinstall /u01
# chmod -R 775 /u01
```

#### 11. Edit bash profiles to set up user environments:

```
# vim /home/oracle/.bash_profile
# Oracle Settings
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=orcl.localdomain
export ORACLE_BASE=/u01/app/oracle
export GRID_HOME=$ORACLE_BASE/product/12.1.0/grid
export DB_HOME=$ORACLE_BASE/product/12.1.0/dbhome_1
export ORACLE_HOME=$DB_HOME
export ORACLE_SID=orcl
export ORACLE_TERM=xterm
export BASE_PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
alias grid_env='. /home/oracle/grid_env'
alias db_env='. /home/oracle/db_env'
```

```
# vim /home/oracle/grid_env
```

```
export ORACLE_SID=+ASM1
export ORACLE_HOME=$GRID_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
```

```
# vim /home/oracle/db_env
```

```
export ORACLE_SID=orcl
export ORACLE_HOME=$DB_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
```

## Configure disks for ASM (OLAP only)

### 1. Edit /etc/scsi\_id.config:

```
# echo "options=-g" > /etc/scsi_id.config
```

### 2. Record the serial IDs for each disk by running `scsi_id /dev/device_id`

### 3. Create /etc/udev/rules.d/99-oracle-asmdevices.rules:

```
# vim /etc/udev/rules.d/99-oracle-asmdevices.rules
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29dbbd2bac74ea0aa589f388847",
    NAME+="oracleasm/data01", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c298d65f5921ec9b9ce1ddfb915c",
    NAME+="oracleasm/data02", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29e67edc3e835800f57463f78bd",
    NAME+="oracleasm/data03", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29d29567a601b785ae99e2d7ade",
    NAME+="oracleasm/data04", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29412bdfbafafb71f38b64e304b",
    NAME+="oracleasm/redo01", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29407694010d5193dfeb5bfddb1",
    NAME+="oracleasm/redo02", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29e89341803fb88fd7d178245d1",
    NAME+="oracleasm/redo03", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c2949482b90b3ff5ba3889f12208",
    NAME+="oracleasm/redo04", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29eac8aa5299869b9e7b82069b8",
    NAME+="oracleasm/temp01", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29d949d2bf33a733242e5ee06fe",
    NAME+="oracleasm/temp02", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c29729713c80efcf00499734478c",
    NAME+="oracleasm/temp03", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd*", SUBSYSTEM=="block", ENV{DEVTYPE}=="disk",
    ENV{ID_SERIAL}=="36000c298bc641566583e7000dfda83fe",
    NAME+="oracleasm/temp04", OWNER="oracle", GROUP="dba", MODE="0660"
```

## Installing Oracle Grid Infrastructure 12c (OLAP only)

### 1. Unzip `linuxamd64_12c_grid_1of2.zip` and `linuxamd64_12c_grid_2of2.zip`

2. Open a terminal to the unzipped database directory.
3. Set the Oracle grid environment.
4. To start the installer, type `./runInstaller`
5. At the software Updates screen, select Skip updates.
6. In the Select Installation Option screen, select Install and Configure Grid Infrastructure for a Standalone Server, and click Next.
7. Choose the language, and click Next.
8. In the Create ASM Disk Group screen, choose the Disk Group Name, change redundancy to External.
9. Select the four disks that you are planning to use for the database, and click Next.
10. In the Specify ASM Password screen, write the passwords for the ASM users, and click Next.
11. Leave the default Operating System Groups, and click Next.
12. Leave the default installation, and click Next.
  - a. Leave the default inventory location, and click Next.
  - b. Under Root script execution select Automatically run configuration scripts and enter root credentials.
  - c. In the Prerequisite Checks screen, make sure that there are no errors.
  - d. In the Summary screen, verify that everything is correct, and click Finish to install Oracle Grid Infrastructure.
  - e. At one point during the installation, the installation prompts you to execute two configuration scripts as root. Follow the instructions to run the scripts.
  - f. At the Finish screen, click Close.
13. To run the ASM Configuration Assistant, type `asmca`.
14. In the ASM Configuration Assistant, click Create.
15. In the Create Disk Group window, name the new disk group `log` choose redundancy External (None), select the four disks for redo logs, and click OK.
16. In the ASM Configuration Assistant, click Create.
17. In the Create Disk Group window, name the new disk group `temp` choose redundancy External (None), select the four disks for the temp tablespace, and click OK.
18. Exit the ASM Configuration Assistant.

### Creating the ASM disk groups (OLAP only)

1. Login to sqlplus:

```
# sqlplus / as sysasm
```

2. Run the following sqlplus commands:

```
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'compatible.asm' = '12.1';
SQL> ALTER DISKGROUP DATA SET ATTRIBUTE 'compatible.rdbms' = '12.1';
SQL> CREATE DISKGROUP REDO external REDUNDANCY disk
'/dev/oracleasm/redo01',
'/dev/oracleasm/redo02',
'/dev/oracleasm/redo03',
'/dev/oracleasm/redo04'
ATTRIBUTE
    'au_size'='1M',
    'compatible.asm' = '12.1',
    'compatible.rdbms' = '12.1';
```

```
SQL> CREATE DISKGROUP TEMP external REDUNDANCY disk
'/dev/oracleasm/temp01',
'/dev/oracleasm/temp02',
'/dev/oracleasm/temp03',
'/dev/oracleasm/temp04'
ATTRIBUTE
    'au_size'='4M',
    'compatible.asm' = '12.1',
    'compatible.rdbms' = '12.1';
```

### Installing Oracle Database 12c

1. Unzip linuxamd64\_12c\_database\_1\_of\_2.zip and linuxamd64\_12c\_database\_2\_of\_2.zip.
2. Open a terminal to the unzipped database directory.
3. Set the Oracle database environment.
4. Run ./runInstaller.sh.
5. Wait for the GUI installer loads.
6. On the Configure Security Updates screen, enter the credentials for My Oracle Support. If you do not have an account, uncheck the box I wish to receive security updates via My Oracle Support, and click Next.
7. At the warning, click Yes.
8. On the Download Software Updates screen, enter the desired update option, and click Next.
9. On the Select Installation Option screen, select Install database software only, and click Next.
10. On the Grid Installation Options screen, select Single instance database installation, and click Next.
11. On the Select Product Languages screen, leave the default setting of English, and click Next.
12. On the Select Database Edition screen, select Enterprise Edition, and click Next.
13. On the Specify Installation Location, leave the defaults, and click Next.
14. On the Create Inventory screen, leave the default settings, and click Next.
15. On the Privileged Operating System groups screen, keep the defaults, and click Next.
16. Allow the prerequisite checker to complete.
17. On the Summary screen, click Install.
18. Once the Execute Configuration scripts prompt appears, ssh into the server as root, and run the following commands:

```
# /home/oracle/app/oraInventory/orainstRoot.sh
# /home/oracle/app/oracle/product/12.1.0/dbhome_1/root.sh
```

19. Return to the prompt, and click OK.
20. Once the installer completes, click Close.

### Create the HammerDB database (OLAP only)

We used the following script to create our HammerDB database:

```
#!/bin/ksh
#####
# create tpch database
#####
echo 300GB database creation started at `date`
```

```

sqlplus /NOLOG <<!
connect / as sysdba
set echo on
set timing on

shutdown abort;

startup pfile=?/dbs/inittpch.ora nomount;
create database
  controlfile reuse
  set default bigfile tablespace
  logfile group 1 ('+REDO/redo01.log') size 10g reuse,
             group 2 ('+REDO/redo02.log') size 10g reuse,
             group 3 ('+REDO/redo03.log') size 10g reuse
  datafile '+DATA/system.dbf' size 2g reuse
  sysaux datafile '+DATA/sysaux.dbf' size 4g reuse
  smallfile undo tablespace ts_undo
  datafile '+DATA/ts_undo01.dbf' size 15g reuse
  default temporary tablespace temp
  tempfile '+TEMP/temp.dbf'
             size 100000m reuse
             extent management local uniform size 10m
  maxdatafiles 2000
  maxinstances 1;
!echo 300GB Database created

!echo 300GB Creating dictionary

set termout off
set echo off
spool /tmp/cat
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catparr.sql;
@?/rdbms/admin/catproc.sql;
connect system/manager
@?/rdbms/admin/utlxplan.sql;
@?/sqlplus/admin/pupbld.sql;
spool off
exit;
!

echo End Database Creation at `date`
*****End Script*****

```

**Add additional tablespace for data:**

```
SQL> create tablespace tpchtab datafile '+DATA/tpchtab.dbf' size 450g reuse
extent management local autoallocate ;
```

### Creating the DVDstore database (OLTP only)

1. Type `dbca`, and press enter to open the Database configuration assistant.
2. At the Database Operation screen select Create Database, and click Next.
3. Under Creation Mode select Advanced Mode, and click Next.
4. At the Select Template screen select General Purpose or Transaction Processing. Click Next
5. Enter a Global database name and the appropriate SID.
6. At the Management Options screen select Configure Enterprise Manager (EM) Database Express. Click Next.
7. At the Database Credentials screen select Use the Same Administrative Password for All Accounts. Enter a password, and click Next.
8. At the Network Configuration screen click Next.
9. At the Storage Configuration screen select File System, and specify the database location.
10. At the Database Options screen click Next.
11. At the Initialization Parameters screen click use Automatic Memory Management.
12. At the Creation Options select Create Database, and click Next.
13. At the summary screen click Finish.
14. Close the Database Configuration Assistant.

### Generate HammerDB data (OLAP only)

We generated the data using two Windows Server 2008 VM clients with HammerDB installed.

1. Download the HammerDB install from [hammerora.sourceforge.net/download.html](http://hammerora.sourceforge.net/download.html)
2. Double-click the executable to install HammerDB on the client.
3. Click Run.
4. Choose English, and click OK.
5. Click Yes on the install prompt.
6. Click Next.
7. Leave the default installation destination, and click Next.
8. Click Next.
9. Check Launch HammerDB, and click Finish.
10. In the HammerDB UI, click Options→Benchmark, and check Oracle and TPC-H. Click OK.
11. Click OK again to confirm the benchmark choice.
12. Expand TPC-H and Schema Build.
13. Double-click Options to open the Build Options menu.
14. For Oracle Service Name, type `<IP_of_TPC-H_Server>:1521/<name_of_database>`
15. Leave the rest of the fields as default.
16. Choose 300 for the Scale Factor, and click OK.
17. Open the Driver Script Options, and set the Degree of Parallelism to 2. Click OK.
18. In the Virtual User Options, check Show Output, Log Output to Temp, and Use Unique Log Name. Click OK.
19. To start the database generation, double-click Build.

## Configuring the DVD Store database (OLTP only)

### Data generation overview

We generated the data using the Install.pl script included with DVD Store version 2.1 (DS2), providing the parameters for our 30GB database size and the database platform on which we ran: Oracle Database. We ran the Install.pl script on each VM. The database schema was also generated by the Install.pl script.

We created VMware snapshots of each VM after we finished creating the databases. Between runs, we restored each VM to the most recent snapshot.

We created additional indexes to improve lookup performance and reduce table scans. Additionally, we modified the login stored procedure to bypass using the temporary tablespace. Finally, we also modified the data generation scripts (see DVD store modifications section below). Specifically, we followed the steps below:

1. We generated the data and created the database and file structure using database creation scripts in the DS2 download. We made size modifications specific to our 30GB database.
2. We created database tables, stored procedures, and objects using the provided DVD Store scripts.
3. We loaded the data we generated into the database using sqlldr and the provided DVD Store load scripts.
4. We created indices, full-text catalogs, primary keys, and foreign keys using our modified database-creation scripts.
5. We created a database user, and mapped this user to the Oracle Database login.
6. We then took a VM snapshot to use as a restore point for resetting the test.

### DVD Store modifications

We made a few modifications to the database creation scripts for DVD store to increase the databases' performance. We modified the files in the following ways:

1. oracleds2\_create\_ind.sql
  - Created a new index on the CUST\_HIST table that included the CUSTOMERID and PROD\_ID columns.
  - Changed the PK\_ORDERS index to a reverse index.
  - Created a new index on the ORDERS table that included the CUSTOMERID column.
  - Changed the PK\_ORDERLINES index to a reverse index.
  - Created a new index on the PRODUCTS table that included the PROD\_ID and COMMON\_PROD\_ID columns.
  - Created a new index on the PRODUCTS table that included the SPECIAL, CATEGORY, and PROD\_ID columns.
  - Changed the IX\_INV\_PROD\_ID index to a unique index and made it the primary key.
  - Created a new index on the REORDER table that included the PROD\_ID column.
2. oracleds2\_create\_sp.sql
  - Removed the "derivedtable" global temporary table creation.
  - Modified the LOGIN procedure to not use the "derivedtable" global temporary table.
3. oracleds2\_create\_tablespaces\_30GB.sql
  - Used five SMALLFILE data files with unlimited maximum size for the CUSTTBS tablespace.
  - Used five SMALLFILE data files with unlimited maximum size for the INDXTBS tablespace.
  - Used five SMALLFILE data files with unlimited maximum size for the ORDERTBS tablespace.
  - Used one SMALLFILE data file with unlimited maximum size for the DS\_MISC tablespace.

## Setting up the client clients

We ran eight DVD Store clients, and two HammerDB clients installed with Windows Server 2008, one targeting each VM of the appropriate type. Each DVDstore client contained the ds2oracle driver executable, and a script that could be run to start the executable with the necessary parameters for the run:

```
c:\ds2\ds2oracledriver.exe --target=10.41.5.223:1521/orcl --db_size=30GB --  
run_time=1000 --n_threads=6 --think_time=.5 --warmup_time=1 --  
detailed_view=Y --csv_output=c:\ds2\output\DS2_client1.csv
```

Each HammerDB client ran HammerDB with the parameters outlined above in the HammerDB data creation section. To run HammerDB, we used RDP to connect to the two clients, loaded the Driver Script into the script editor, clicked on the Create Users button, and started the run.

## Running the tests

We positioned our VMs in the following manner across the four Cisco B200 M3 hosts:

- Host 1
  - 1 x OLAP VM
  - 2 x OLTP VMs
- Host 2
  - 2 x OLTP VMs
- Host 3
  - 1 x OLAP VM
  - 2 x OLTP VMs
- Host 4
  - 2 x OLTP VMs

With the use of various automation scripts, we performed three runs: a baseline run without vFRC enabled and without performing a vMotion task; a run with vFRC enabled, but with no vMotion task; and a run with vFRC enabled and performing a vMotion task roughly halfway through a TPC-H run.

1. Clean up prior outputs from the host system and all client driver systems.
2. Reboot the Cisco blades and all client systems.
3. Reset all server VMs to the latest snapshots.
4. Restart the Oracle instance.
5. Start the esxstop gathering script.
6. If using vFRC, start the vFRC data gathering scripts.
7. Start the DVD Store script.
8. Start the two HammerDB TPC-H runs.
9. If performing a vMotion task, migrate the VMs on Host 1 about 1 hour, 23 minutes (the halfway mark in first vFRC run) into the TPC-H run.
10. When the TPC-H tasks finish on both VMs, sftop the esxstop, DVD Store, and cache scripts.
11. Gather all the output files from esxstop, esxcli cache, DVD Store clients, and TPC-H clients.

## APPENDIX D – SCRIPTS WE USED

We used the following scripts to help automate the run process.

### DVD Store Scripts

#### Client Script

```
c:\ds2\ds2oracledriver.exe --target=10.41.5.223:1521/orcl --db_size=30GB --
run_time=1000 --n_threads=6 --think_time=.5 --warmup_time=1 --
detailed_view=Y --csv_output=c:\ds2\output\DS2_client1.csv
```

#### Controller start DVD Store script

```
start psexec \\10.41.5.203 -u Administrator -p Password1 C:\ds2\runds2.bat
start psexec \\10.41.5.204 -u Administrator -p Password1 C:\ds2\runds2.bat
start psexec \\10.41.5.205 -u Administrator -p Password1 C:\ds2\runds2.bat
start psexec \\10.41.5.206 -u Administrator -p Password1 C:\ds2\runds2.bat
start psexec \\10.41.5.207 -u Administrator -p Password1 C:\ds2\runds2.bat
start psexec \\10.41.5.208 -u Administrator -p Password1 C:\ds2\runds2.bat
start psexec \\10.41.5.209 -u Administrator -p Password1 C:\ds2\runds2.bat
start psexec \\10.41.5.210 -u Administrator -p Password1 C:\ds2\runds2.bat
```

#### Controller end DVD Store script

```
taskkill /s 10.41.5.203 /u Administrator /p Password1 /im ds2oracledriver.exe /f
taskkill /s 10.41.5.204 /u Administrator /p Password1 /im ds2oracledriver.exe /f
taskkill /s 10.41.5.205 /u Administrator /p Password1 /im ds2oracledriver.exe /f
taskkill /s 10.41.5.206 /u Administrator /p Password1 /im ds2oracledriver.exe /f
taskkill /s 10.41.5.207 /u Administrator /p Password1 /im ds2oracledriver.exe /f
taskkill /s 10.41.5.208 /u Administrator /p Password1 /im ds2oracledriver.exe /f
taskkill /s 10.41.5.209 /u Administrator /p Password1 /im ds2oracledriver.exe /f
taskkill /s 10.41.5.210 /u Administrator /p Password1 /im ds2oracledriver.exe /f
```

### Esxtop data gathering scripts

We created one script for each host, and used command prompts to start each script. The locations used in the scripts are specific to our testing and will be different when run in other tests.

```
plink.exe 10.41.5.241 -l root -pw Password1 esxtop -b -n $1 -d 5 >
C:\Valenti_Results\esxtop\10.41.5.241.csv &
plink.exe 10.41.5.242 -l root -pw Password1 esxtop -b -n $1 -d 5 >
C:\Valenti_Results\esxtop\10.41.5.242.csv &
plink.exe 10.41.5.243 -l root -pw Password1 esxtop -b -n $1 -d 5 >
C:\Valenti_Results\esxtop\10.41.5.243.csv &
plink.exe 10.41.5.244 -l root -pw Password1 esxtop -b -n $1 -d 5 >
C:\Valenti_Results\esxtop\10.41.5.244.csv &
```

### Cache data gathering scripts

We stored these scripts on the storage LUNs that were not being used to hold the data VMDKs, and ran them with command lines on each host that held cache. With each run, we had to use the esxcli storage vflash cache list command to determine the cache names on each host. We also used the command after the vMotion task to determine

the cache names on the target host once the OLAP VM moved. We edited the scripts with the correct task names, ran the reset script, then started the gathering scripts that provided cache data every five minutes.

#### Cache gathering script

```
for i in $(seq 1 1 300)
do
    esxcli storage vflash cache stats get -c vfc-1457171792-OLAP-002_1-000003 >>
        /vmfs/volumes/53136407-e8f3c456-b35c-0025b510002f/cache_results3/OLAP-
        002_cache1.log
    esxcli storage vflash cache stats get -c vfc-1457171792-OLAP-002_2-000003 >>
        /vmfs/volumes/53136407-e8f3c456-b35c-0025b510002f/cache_results3/OLAP-
        002_cache2.log
    esxcli storage vflash cache stats get -c vfc-1457171792-OLAP-002_3-000003 >>
        /vmfs/volumes/53136407-e8f3c456-b35c-0025b510002f/cache_results3/OLAP-
        002_cache3.log
    esxcli storage vflash cache stats get -c vfc-1457171792-OLAP-002_4-000003 >>
        /vmfs/volumes/53136407-e8f3c456-b35c-0025b510002f/cache_results3/OLAP-
        002_cache4.log
    sleep 300
done
```

#### Cache reset script (the reset command does not generate any output, so logging was not required)

```
for i in $(seq 1 1 300)
do
    esxcli storage vflash cache stats reset -c vfc-1457171792-OLAP-002_1-000003
    esxcli storage vflash cache stats reset -c vfc-1457171792-OLAP-002_2-000003
    esxcli storage vflash cache stats reset -c vfc-1457171792-OLAP-002_3-000003
    esxcli storage vflash cache stats reset -c vfc-1457171792-OLAP-002_4-000003
    sleep 300
done
```

## ABOUT PRINCIPLED TECHNOLOGIES



Principled Technologies, Inc.  
1007 Slater Road, Suite 300  
Durham, NC, 27703  
[www.principledtechnologies.com](http://www.principledtechnologies.com)

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help our clients assess how it will fare against its competition, its performance, its market readiness, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.

---

Principled Technologies is a registered trademark of Principled Technologies, Inc.  
All other product names are the trademarks of their respective owners.

---

#### Disclaimer of Warranties; Limitation of Liability:

PRINCIPLED TECHNOLOGIES, INC. HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, PRINCIPLED TECHNOLOGIES, INC. SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT PRINCIPLED TECHNOLOGIES, INC., ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC. BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL PRINCIPLED TECHNOLOGIES, INC.'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH PRINCIPLED TECHNOLOGIES, INC.'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.