CONSOLIDATING ORACLE DATABASE SERVERS ON THE DELL POWEREDGE R930

DELL™ POWEREDGE™ R930 SERVER

CONSOLIDATED 3 LEGACY SERVERS

*as reported by HammerDB

4.4X
the database transactions per minute*
on an Oracle® Database 12c workload versus a legacy server

67%
less rack space required
versus the three legacy servers it would replace

25%
savings on Oracle Database12c licensing
versus the cost of licensing three legacy servers

POWERED BY THE INTEL® XEON® PROCESSOR E7-8800 V3 FAMILY

Your databases are at the core of your business. Running them on older servers can make it difficult to deliver consistent top-notch performance to customers, and can spin your IT budget out of control as the number of machines expand. By consolidating databases onto a new platform, you can increase overall database performance, reduce needed rack space, and potentially reduce costs associated with storing, powering, and managing your infrastructure.

In the Principled Technologies data center, we compared a new Dell PowerEdge R930 server to an older generation Dell PowerEdge R910. Powered by the Intel Xeon processor E7 v3 family, the PowerEdge R930 with 12 virtual machines delivered 4.4 times the database transactions per minute of a Dell PowerEdge R910 with four virtual machines, as reported by HammerDB. This means it could consolidate the workload of three older PowerEdge R910 servers while delivering an average of 47.1 percent more performance per VM. With this level of consolidation, the Dell PowerEdge R930 could save 67 percent in rack space and 25 percent in Oracle Database 12*c* licenses compared to legacy servers doing nearly the same amount of work.

JUNE 2015
A PRINCIPLED TECHNOLOGIES TEST REPORT
Commissioned by Dell Inc.

CONSOLIDATE YOUR WORKLOADS WITH BETTER PERFORMANCE

Doing more with less is a surefire way to realize big savings in business. The same holds true with the hardware in your data center. By consolidating older database servers onto a newer, more powerful server that utilizes the latest technology, you can do the same amount of work in less space. By reducing the number of servers you have to store, rack, cable, power, cool, license, and manage, you can maximize the efficiency of the racks that populate your data center. In addition to the consolidation savings, upgrading your server for better database performance can translate to a better user experience and more orders flowing to your business.

The latest in the line of four-socket servers that includes the 11th generation PowerEdge R910 server, the Dell PowerEdge R930 is a 4U server that features the newest Intel Xeon processor E7 v3 families, 96 DIMM slots, and 24 drive bays. To show the benefits that the Dell PowerEdge R930 can provide, we tested the performance of virtualized Oracle Database 12*c* instances on the servers to find how many database instances the new 13G PowerEdge server could support.

For information about the Dell PowerEdge R930 and our other test components, see Appendix A. For detailed system configuration, see Appendix B. For step-by-step testing details, see Appendix C.

WHAT WE FOUND

By utilizing the latest technology, the Dell PowerEdge R930 server was able to run more virtualized database instances and significantly improved overall database performance compared to the legacy server, all in less physical space.

As Figure 1 shows, the Dell PowerEdge with Intel Xeon processors E7 v3 R930 delivered three times the number of virtual machines running Oracle Database 12*c* than the legacy server did, which means that you could replace three older servers with a single new PowerEdge R930.

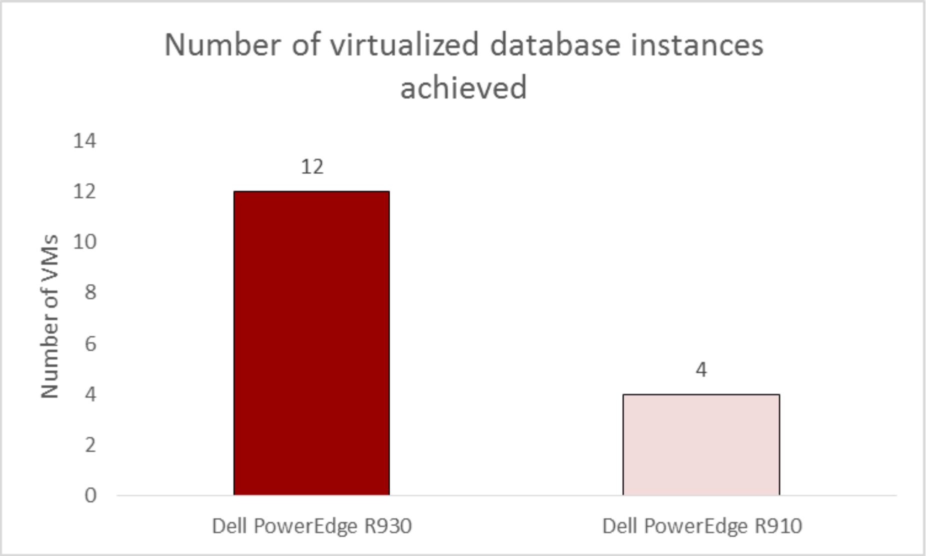

Figure 1: The Dell PowerEdge R930 delivered three times the number of virtual machines of the legacy server. Higher numbers are better.

Not only did the Dell PowerEdge R930 handle three times the number of VMs, it got more performance from each database virtual machines as well. The Dell PowerEdge R930 achieved a total of 4.4 times the transactions per minute of the legacy server, as reported by HammerDB. More database transactions can mean orders come in quickly to help your business do more, and each of the 12 VMs on the PowerEdge R930 delivered an average of 47.1 percent more performance than each of the VMs on the legacy server. Figure 2 shows the relative overall database performance of each server. We normalized performance to the Dell PowerEdge R910 because Oracle does not permit publishing specific benchmark results.

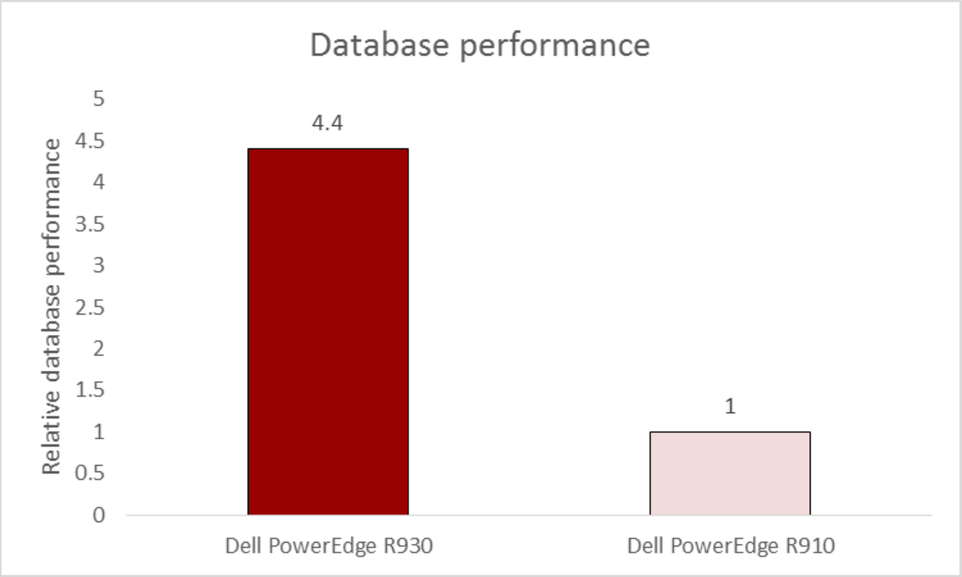

Figure 2: The Dell PowerEdge R930 achieved 4.4 times the database performance of the legacy server, running 12 VMs instead of just 4 VMs. Higher numbers are better.

Both the PowerEdge R930 and legacy server take up 4U of rack space. By consolidating three legacy servers, the Dell PowerEdge R930 server with Intel Xeon processors E7 v3 can do the work of three servers in 67 percent less space, meaning potential savings in cable management, power usage and management time. Figure 3 shows the rack space savings of the Dell PowerEdge R930 compared to the legacy server.

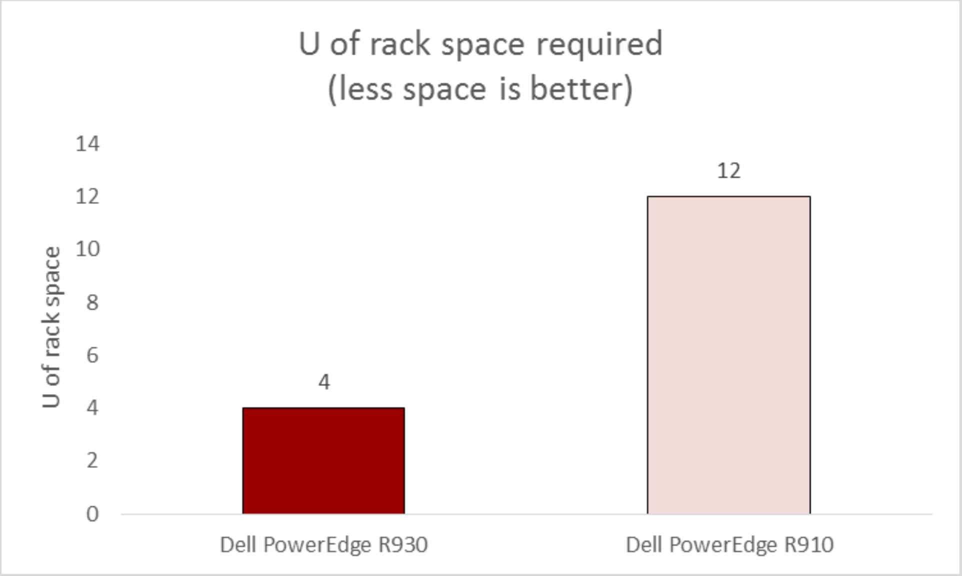

Figure 3: The Dell PowerEdge R930 was able to achieve the same or better performance in 67% less rack space. Lower numbers are better.

Consolidating your servers into a newer, more powerful server can also reduce licensing costs. Because one Dell PowerEdge R930 with Intel Xeon processors E7-8890 v3 and 72 cores was able to do the work of three Dell PowerEdge R910 servers with 32 cores each, you could save $946,000 on Oracle Database 12*c* licenses over three years, cutting licensing costs by 25 percent (see Figure 4).

	Dell PowerEdgeR930	Dell PowerEdge R910
Cores per server	72	32
Number servers in solution (3:1 consolidation)	1	3
Number cores per solution	72	96
Core factor	0.5	0.5
Number licenses	36	48
License cost	$2,838,600.00	$3,784,800.00
Savings	$946,200.00	
Savings (%)	25%	

Figure 4: Oracle Database 12c licensing cost comparison.[1]

[1] Costs based on an Oracle Database Enterprise Edition license cost of $47,500 and a Software Update License and Support cost (per year) of $10,450 for a three-year license cost of $78,850.

CONCLUSION

Moving your legacy database workloads to the Dell PowerEdge R930 can help you realize the benefits of consolidation, which can include savings in management costs, power usage, and cable management costs. More importantly, the licensing costs of the database application itself may be reduced by the consolidation effort. In addition to these benefits, greater database transactions per minute can keep your orders flowing smoothly.

We found that the Dell PowerEdge R930, powered by the Intel Xeon processor E7 v3 series, could consolidate three legacy servers running four Oracle Database 12*c* VMs each. The Dell PowerEdge R930 outperformed the legacy server with 4.4 times the overall database performance, delivering an average of 47.1 percent more performance per VM. By consolidating that many legacy servers, you can save up to 67 percent in rack space, 25 percent in database licenses, and even reduce other operating costs to improve your bottom line.

Prices from https://shop.oracle.com/pls/ostore/product?p1=OracleDatabaseEnterpriseEdition&p2=&p3=&p4=&intcmp=ocom_oracledatabaseenterpriseedition.

APPENDIX A – ABOUT THE COMPONENTS

About the Dell PowerEdge R930

The Dell PowerEdge R930 is part of Dell 13G offerings and is currently the fastest four-socket 4U server. Designed to provide expanding scalable performance for large enterprises, it supports up to:

- 96 DIMMs of DDR4 memory
- 24 internal drives with the option for 8 PCIe® Express Flash drives (with the optional PCIe backplane)
- 12Gb/s SAS drives

It also offers Dual PERC option, PERC9 (H730P), Fluid Cache for SAN capability, and a number of built-in RAS features for high reliability, such as Fault Resilient memory and Intel Run Sure technology. To learn more about the PowerEdge R930, visit www.dell.com/learn/us/en/04/campaigns/poweredge-13g-server.

About the Intel Xeon processor E7-8800 v3 and E7-4800 v3 families

Intel designed the new Intel Xeon processor E7-8800 v3 and E7-4800 v3 families to support mission-critical, high-performance workloads by featuring up to 18 cores and 36 threads per socket to provide significant jumps in performance from previous releases. The Intel Xeon processor E7-8800 v3 and E7-4800 v3 families support up to 1.5TB DDR4 memory per socket, support up to 24 DDR4 DIMMs per socket, and support up to 2,133MHz DDR4 speeds to improve performance and increase scalability.

The Intel Xeon processor E7-8800 v3 and E7-4800 v3 families support all the previous reliability, availability, and serviceability features of previous processor releases to support critical workloads. With Intel Run Sure technology, these processors add new RAS features, including eMCA Gen 2, MCA Recovery – Execution Path, MCA IO, and PCIe Live Error Recovery. For more information about the Intel Xeon processor E7 v3 product family, visit www.intel.com.

About the benchmark we used – HammerDB

HammerDB is an open-source benchmark tool that tests the database performance of many databases, including Oracle Database, Microsoft® SQL Server®, PostgreSQL®, MySQL™, and more. The benchmark includes built-in workloads derived from industry-standard benchmarks, such as a transactional (derived from TPC-C-) workload and a data warehouse (TPC-H-like) workload. For this study, we used the transactional workload. The TPC-C-derived HammerDB workload is derived from TPC-C and as such is not comparable to published TPC-C results. For more information, visit www.hammerora.sourceforge.net.

APPENDIX B – SYSTEM CONFIGURATION INFORMATION

Figure 5 provides detailed configuration information for the test systems.

System	Dell PowerEdge R930	Dell PowerEdge R910
Power supplies		
Total number	4	4
Vendor and model number	Dell 0GDPF3	Dell 0TCVRRA00
Wattage of each (W)	1,100	1,100
Cooling fans		
Total number	6	6
Vendor and model number	Nidec® UltraFlo V12C12BS1M3	Delta Electronics PFC1212DE
Dimensions (h × w) of each	5″ × 5″	5.25″ × 5.25″
Volts	12	12
Amps	2.31	4.80
General		
Number of processor packages	4	4
Number of cores per processor	18	8
Number of hardware threads per core	2	2
System power management policy	Performance per watt (OS)	Performance per watt (OS)
CPU		
Vendor	Intel	Intel
Name	Xeon	Xeon
Model number	E7-8890 v3	X7560
Socket type	FCLGA2011	LGA1567
Core frequency (GHz)	2.50	2.27
Bus frequency	9.6 GT/s	6.4 GT/s
L1 cache	32 KB + 32 KB (per core)	32 KB + 32 KB (per core)
L2 cache	256 KB (per core)	256 KB (per core)
L3 cache	45 MB	24 MB
Platform		
Vendor and model number	Dell PowerEdge R930	Dell PowerEdge R910
Motherboard model number	0Y4CNCX19	0JRJM9
BIOS name and version	0.0.23	2.10.0
BIOS settings	Performance per watt (OS)	Performance per watt (OS)
Memory module(s)		
Total RAM in system (GB)	256	128
Vendor and model number	Samsung® M393A1G43DB0-CPB	Samsung M393B1K70BH1-CH9
Type	PC4-2133	PC3-10600
Speed (MHz)	2,133	1,333
Speed running in the system (MHz)	2,133	1,067
Timing/Latency (tCL-tRCD-tRP-tRASmin)	15-15-15-33	9-9-9-36
Size (GB)	8	8
Number of RAM module(s)	32	16

System	Dell PowerEdge R930	Dell PowerEdge R910
Chip organization	Double-sided	Double-sided
Rank	Dual	Dual
Operating system		
Name	Red Hat® Enterprise Linux® 6.5	Red Hat Enterprise Linux 6.5
Build number	2.6.32-431.el6.x86_64	2.6.32-431.el6.x86_64
File system	ext4	ext4
Language	English	English
RAID controller		
Vendor and model number	PERC H730P	PERC H700 Integrated
Firmware version	25.2.2-0004	12.10.6-0001
Driver version	6.700.06.00-rh1	4.31.1.64
Cache size (MB)	2,048	512
Solid-state drives		
Vendor and model number	SanDisk LT0200MO	Pliant LB206M
Number of drives	12	8
Size (GB)	200	200
Type	SAS	SAS
Hard drives #1		
Vendor and model number	Seagate® ST9900805SS	Toshiba MBF2600RC
Number of drives	4	2
Size (GB)	900	600
RPM	10k	10k
Type	SAS	SAS
Hard drives #2		
Vendor and model number	Seagate ST9300653SS	Seagate ST300MM0006
Number of drives	8	6
Size (GB)	300	300
RPM	15k	10k
Type	SAS	SAS
Ethernet adapters		
Vendor and model number	Intel Gigabit 4P i350-t rNDC	Broadcom® BCM5709C
Type	Gigabit Ethernet	Gigabit Ethernet
Driver	12.11.97.9900 2/12/2015	7.12.2.0 11/20/2014

Figure 5: System configuration information for the test systems.

APPENDIX C – HOW WE TESTED

Overview of our configuration

We configured each server to have single RAID10 volumes for the OS and Log virtual disks, and RAID1 SSD pairs for the Data virtual disks. The Dell PowerEdge R930 had four HDDs for the hypervisor and OS virtual disks, 12 SSDs for the Data virtual disks, and eight HDDs for the Log virtual disks. The Dell PowerEdge R910 had two HDDs, eight SSDs, and six HDDs respectively. We turned off the PERC read and write caching for the Data SSD volumes and we turned off read caching on the Log HDD volumes. On the Data volumes, we divided the RAID into portions: six groups of two disks each on the Dell PowerEdge R930, and four groups of two disks each on the Dell PowerEdge R910.

We used Red Hat Enterprise Linux 6.5 with the KVM host package as our hypervisor. Our guest OS was also Red Hat 6.5, and our database software was Oracle Database 12*c*.

We created one virtualized instance of Oracle Database 12*c* on the Dell PowerEdge R910 to be our base VM and cloned the base VM until we had four VMs on the legacy Dell PowerEdge R910. We then migrated the base VM to the Dell PowerEdge R930 and cloned it until we had 12 VMs on the upgraded server.

We restored the database to a fresh copy and rebooted all guests and the host before each test run. We performed the test three times and reported the median result.

Configuring the Oracle Virtual Machine environment

We set the power profile Performance-per-watt (OS) so that we could use Red Hat’s power management features. We left the other BIOS settings as default.

Installing RHEL 6.5 with KVM on the hypervisor host

Perform the following steps on both hypervisor hosts: The Dell PowerEdge R910 and the Dell PowerEdge R930.

1. Boot to the Red Hat Enterprise Linux 6.5 installation disc.
2. Select Install or upgrade an existing system and press Enter.
3. Skip the installation media integrity check and enter the installation process.
4. At the RHEL 6.5 splash screen click Next.
5. Select your language and click Next.
6. Select your keyboard layout and click Next.
7. Select Basic Storage Devices and click Next.
8. Click Yes, discard any data.
9. Give the hypervisor host a hostname.
10. Click Configure Network, and select the NIC to be used for testing. Assign the NIC a static IPv4 address and subnet mask. Click OK.
11. Click Next.
12. Select the time zone to be used and click Next.
13. Provide a password for the root user and click Next.
14. Select Use All Space and click Next.
15. Click Write changes to disk.
16. Select Virtualization Host and click Next.
17. When the OS has finished installing click Reboot.

Configuring the Dell PowerEdge R930

Perform these steps on the Dell PowerEdge R930 only.

1. Install the following packages on the server:

```
yum install -y time tuned numactl acpid cpuspeed wget vim nfs-utils openssh-
clients man unzip numactl ipmitool OpenIPMI sysstat
```

2. Disable SELinux and the firewall:

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

3. Edit /etc/fstab:

```
CHANGE:
/dev/mapper/vg_r930-lv_root /                       ext4    defaults        1 1
/dev/mapper/vg_r930-lv_home /home                   ext4    defaults        1 2

TO:
/dev/mapper/vg_r930-lv_root /                       ext4
defaults,nobarrier,noatime,nodiratime        1 1
/dev/mapper/vg_r930-lv_home /home                   ext4
defaults,nobarrier,noatime,nodiratime        1 2

ADD:
hugetlbfs     /dev/hugepages          hugetlbfs       mode=1770,gid=107       0 0
```

4. Add the following to `/etc/libvirt/qemu.conf`:

```
hugetlbfs_mount = "/dev/hugepages"
```

5. Add the following to `/etc/sysctl.conf`:

```
vm.hugetlb_shm_group = 107
vm.nr_hugepages = 122976
```

6. Run the following commands to sync the time to your NTP server:

```
service ntpd stop
sed -i '/server .*iburst/d' /etc/ntp.conf
echo 'server <NTP SERVER HOSTNAME> iburst prefer' >> /etc/ntp.conf
service ntpd start
chkconfig ntpd on
```

7. Run the following commands to create the LVM volumes and storage pools:

```
pvcreate /dev/sd[b-h]

pvs -o+pe_start
  PV         VG       Fmt  Attr PSize   PFree   1st PE
  /dev/sda2  vg_r930  lvm2 a--    1.64t      0    1.00m
  /dev/sdb            lvm2 a--  185.75g 185.75g   1.00m
  /dev/sdc            lvm2 a--    1.09t   1.09t   1.00m
  /dev/sdd            lvm2 a--  185.75g 185.75g   1.00m
  /dev/sde            lvm2 a--  185.75g 185.75g   1.00m
  /dev/sdf            lvm2 a--  185.75g 185.75g   1.00m
  /dev/sdg            lvm2 a--  185.75g 185.75g   1.00m
  /dev/sdh            lvm2 a--  185.75g 185.75g   1.00m

vgcreate logs  /dev/sdc
vgcreate data1 /dev/sdb
vgcreate data2 /dev/sdd
vgcreate data3 /dev/sde
vgcreate data4 /dev/sdf
vgcreate data5 /dev/sdg
vgcreate data6 /dev/sdh

virsh pool-define-as --name logs --type logical --target /dev/logs
virsh pool-start --pool logs
virsh pool-autostart --pool logs
virsh pool-define-as --name data1 --type logical --target /dev/data1
virsh pool-start --pool data1
virsh pool-autostart --pool data1
virsh pool-define-as --name data2 --type logical --target /dev/data2
virsh pool-start --pool data2
virsh pool-autostart --pool data2
virsh pool-define-as --name data3 --type logical --target /dev/data3
virsh pool-start --pool data3
virsh pool-autostart --pool data3
virsh pool-define-as --name data4 --type logical --target /dev/data4
virsh pool-start --pool data4
virsh pool-autostart --pool data4
virsh pool-define-as --name data5 --type logical --target /dev/data5
virsh pool-start --pool data5
virsh pool-autostart --pool data5
virsh pool-define-as --name data6 --type logical --target /dev/data6
virsh pool-start --pool data6
virsh pool-autostart --pool data6
```

8. Apply the Tuned virtual host profile:

```
tuned-adm profile virtual-host
```

9. Reboot the hypervisor host.

Configuring the Dell PowerEdge R910

Perform these steps on the Dell PowerEdge R910 only.

1. Install the following packages on the server:

```
yum install -y time tuned numactl acpid cpuspeed wget vim nfs-utils openssh-
clients man unzip numactl ipmitool OpenIPMI sysstat
```

2. Disable SELinux and the firewall:

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

3. Edit /etc/fstab:

```
CHANGE:
/dev/mapper/vg_r910-lv_root /                       ext4    defaults        1 1
/dev/mapper/vg_r910-lv_home /home                   ext4    defaults        1 2

TO:
/dev/mapper/vg_r910-lv_root /                       ext4
defaults,nobarrier,noatime,nodiratime        1 1
/dev/mapper/vg_r910-lv_home /home                   ext4
defaults,nobarrier,noatime,nodiratime        1 2

ADD:
hugetlbfs     /dev/hugepages          hugetlbfs       mode=1770,gid=107       0 0
```

4. Add the following to /etc/libvirt/qemu.conf:

```
hugetlbfs_mount = "/dev/hugepages"
```

5. Add the following to /etc/sysctl.conf:

```
vm.hugetlb_shm_group = 107
vm.nr_hugepages = 40992
```

6. Run the following commands to sync the time to your NTP server:

```
service ntpd stop
sed -i '/server .*iburst/d' /etc/ntp.conf
echo 'server <NTP SERVER HOSTNAME> iburst prefer' >> /etc/ntp.conf
service ntpd start
chkconfig ntpd on
```

7. Run the following commands to create the LVM volumes:

```
pvcreate /dev/sd[b-f]

pvs -o+pe_start
  PV         VG        Fmt  Attr PSize   PFree   1st PE
  /dev/sda2  vg_r910   lvm2 a--  837.26g 383.26g   1.00m
  /dev/sdb             lvm2 a--  836.62g 836.62g   1.00m
  /dev/sdc             lvm2 a--  185.75g 185.75g   1.00m
  /dev/sdd             lvm2 a--  185.75g 185.75g   1.00m
  /dev/sde             lvm2 a--  185.75g 185.75g   1.00m
  /dev/sdf             lvm2 a--  185.75g 185.75g   1.00m

vgcreate logs  /dev/sdb
vgcreate data1 /dev/sdc
vgcreate data2 /dev/sdd
vgcreate data3 /dev/sde
vgcreate data4 /dev/sdf

virsh pool-define-as --name logs --type logical --target /dev/logs
virsh pool-start --pool logs
virsh pool-autostart --pool logs
virsh pool-define-as --name data1 --type logical --target /dev/data1
virsh pool-start --pool data1
virsh pool-autostart --pool data1
virsh pool-define-as --name data2 --type logical --target /dev/data2
virsh pool-start --pool data2
virsh pool-autostart --pool data2
virsh pool-define-as --name data3 --type logical --target /dev/data3
virsh pool-start --pool data3
virsh pool-autostart --pool data3
virsh pool-define-as --name data4 --type logical --target /dev/data4
virsh pool-start --pool data4
virsh pool-autostart --pool data4
```

8. Apply the Tuned virtual host profile:

```
tuned-adm profile virtual-host
```

9. Reboot the hypervisor host.

Installing KVM Manager 3.3.1

We first installed RHEL 6.5 with the Workstation package on an auxiliary server and performed the following post-configuration tasks:

1. Disable SELinux and the firewall:

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
service ip6tables stop
```

```
chkconfig iptables off
chkconfig ip6tables off
```

2. Modify the file /etc/hosts to include the IP addresses and hostnames of the VM servers and VMs (yet to be created).

Configuring the VM environment in KVM Manager Web-console

Perform the following steps for each hypervisor.

1. Log onto the KVM Manager server and open Virtual Machine Manager.
2. Click File→Add Connection.
3. Choose QEMU/KVM as the hypervisor.
4. Leave the rest of the settings as default and click Connect.
5. Right-click the server connection and click Details.
6. Click the Virtual Networks tab.
7. Remove the existing virtual network and click the + button to create a new virtual network.
8. Click Forward.
9. Give the virtual network a name and click Forward.
10. Give the virtual network an IP address space and click Forward.
11. Uncheck the Enable DHCP checkbox and click Forward.
12. Select the Forwarding to physical network radio button and select the adapter that is connected to the test network. Select NAT as the forwarding method and click Forward.
13. Click Finish.

Creating VMs for RHEL 6.5 and Oracle Database 12*c*

We configured each VM with one virtual disk on our OS repository running RHEL 6.5 with Oracle Database 12*c*, four (on the R910) and six (on the R930) virtual disks on our Data RAID1 pairs presented to Oracle ASM, and one virtual disk on our Log repository presented to Oracle ASM for redo logs. Below are the detailed steps we used to configure the software.

Creating the first virtual machine

1. On the Servers and VMs tab, click the Create Virtual Machine button.
2. Select Create a new VM, and click Next.
3. Enter a name for the VM.
4. Select the Dell PowerEdge R910 Server.
5. Leave the Local Install media (ISO image or CDROM) radio button selected and click Forward.
6. Browse to the Red Hat Enterprise Linux 6.5 ISO to provide the VM with installation media.

14. Select Red Hat Enterprise Linux 6 for the version.

7. Click Forward.
8. For Memory (RAM), enter `20480`.
9. For CPUs, enter `8`.

15. Click Finish.

Creating virtual disks for the VMs

Follow these steps to create the OS virtual disk on the base VM. Then use the same steps to create the following additional disks: A single 40GB virtual disk for logs, six 10GB virtual disks for data on the R930, and four 15GB virtual disks for data on the R910.

1. In Virtual Machine Manager, right-click the base VM and click Open.
2. Click the Information icon to access the VM details.
3. Click Add Hardware.
4. Click Browse.
5. Select the appropriate storage pool for this virtual disk.
6. Click New Volume.
7. Give the volume a name that is compatible with the XML file given below.
8. Give the volume one of the sizes listed above.
9. Click Finish.
10. Click Choose Volume.
11. Select Virtio Disk from the Device Type drop-down menu.
12. Select none from the Cache Mode drop-down menu.
13. Select raw from the Storage Format drop-down menu.
14. Click Finish.
15. Select the newly-created virtual disk and expand the Performance options sub-section.
16. Change the IO mode to native.
17. Repeat these steps for the remaining virtual disks.

VM XML configuration on the R930

For VMs on the Dell PowerEdge R930, replace the existing XML with the following, and restart the libvirt service to allow it to take effect. For each new VM, edit the VM name (in the <name> and <disk> tags) and <vcpu> tags. For the <vcpu> tag, increment each processor by one up until the fourth VM, and then start over with the fifth VM. This splits the VMs so that three are on each of the server's four NUMA nodes.

```
<domain type='kvm'>
  <name>orcl01</name>
  <uuid>343e65cd-252c-98c6-3228-51d3c5fba594</uuid>
  <memory unit='KiB'>20971520</memory>
  <currentMemory unit='KiB'>20971520</currentMemory>
  <memoryBacking>
    <hugepages/>
    <nosharepages/>
    <locked/>
  </memoryBacking>
  <vcpu placement='static'
cpuset='0,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60,64,68,72,76,80,84,88,92,96,1
00,104,108,112,116,120,124,128,132,136,140'>8</vcpu>
  <numatune>
    <memory mode='strict' nodeset='0'/>
  </numatune>
  <os>
    <type arch='x86_64' machine='rhel6.5.0'>hvm</type>
    <boot dev='hd'/>
    <boot dev='cdrom'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <cpu mode='host-passthrough'>
  </cpu>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/vg_kvmhost/orcl01-os'/>
      <target dev='vda' bus='virtio'/>
      <serial>os</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/logs/orcl01-logs'/>
      <target dev='vdb' bus='virtio'/>
      <serial>logs</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data1/orcl01-data1'/>
      <target dev='vdd' bus='virtio'/>
      <serial>data1</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data2/orcl01-data2'/>
      <target dev='vde' bus='virtio'/>
      <serial>data2</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data3/orcl01-data3'/>
      <target dev='vdf' bus='virtio'/>
      <serial>data3</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0a' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data4/orcl01-data4'/>
      <target dev='vdg' bus='virtio'/>
      <serial>data4</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0b' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data5/orcl01-data5'/>
      <target dev='vdh' bus='virtio'/>
      <serial>data5</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0c' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data6/orcl01-data6'/>
      <target dev='vdi' bus='virtio'/>
      <serial>data6</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0d' function='0x0'/>
    </disk>
    <controller type='usb' index='0'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
    </controller>
    <interface type='bridge'>
      <mac address='52:54:00:83:0f:1e'/>
      <source bridge='br0'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <serial type='pty'>
      <target port='0'/>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <input type='tablet' bus='usb'/>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='-1' autoport='yes'/>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </memballoon>
  </devices>
</domain>
```

VM XML configuration on the R910

For VMs on the Dell PowerEdge R910, replace the existing XML with the following, and restart the libvirt service to allow it to take effect. For each new VM, edit the VM name (in the <name> and <disk> tags) and <vcpu> tags. For the <vcpu> tag, increment each processor by one up until the fourth VM, so that each VM has its own NUMA node.

```
<domain type='kvm'>
  <name>orcl01</name>
  <uuid>339e7f7b-33ff-bcae-7ce8-48a580d4f78a</uuid>
  <memory unit='KiB'>20971520</memory>
  <currentMemory unit='KiB'>20971520</currentMemory>
  <memoryBacking>
    <hugepages/>
    <nosharepages/>
    <locked/>
  </memoryBacking>
  <vcpu placement='static'
cpuset='0,4,8,12,16,20,24,28,32,36,40,44,48,52,56,60'>10</vcpu>
  <numatune>
    <memory mode='strict' nodeset='0'/>
  </numatune>
  <os>
    <type arch='x86_64' machine='rhel6.5.0'>hvm</type>
    <boot dev='hd'/>
    <boot dev='cdrom'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <cpu mode='host-passthrough'>
  </cpu>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/vg_legacyr910/orcl01-os'/>
      <target dev='vda' bus='virtio'/>
      <serial>os</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/logs/orcl01-logs'/>
      <target dev='vdb' bus='virtio'/>
      <serial>logs</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data1/orcl01-data1'/>
      <target dev='vdd' bus='virtio'/>
      <serial>data1</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x08' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data2/orcl01-data2'/>
      <target dev='vde' bus='virtio'/>
      <serial>data2</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data3/orcl01-data3'/>
      <target dev='vdf' bus='virtio'/>
      <serial>data3</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0a' function='0x0'/>
    </disk>
    <disk type='block' device='disk'>
      <driver name='qemu' type='raw' cache='none' io='native'/>
      <source dev='/dev/data4/orcl01-data4'/>
      <target dev='vdg' bus='virtio'/>
      <serial>data4</serial>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0b' function='0x0'/>
    </disk>
    <controller type='usb' index='0'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2'/>
    </controller>
    <interface type='bridge'>
      <mac address='52:54:00:6c:79:6c'/>
      <source bridge='br0'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <serial type='pty'>
      <target port='0'/>
    </serial>
    <console type='pty'>
      <target type='serial' port='0'/>
    </console>
    <input type='tablet' bus='usb'/>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='-1' autoport='yes'/>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </memballoon>
  </devices>
</domain>
```

Installing Red Hat Enterprise Linux 6.5

We used the same steps as above to install RHEL 6.5 onto the VM, but we did not select the Virtualization Host package.

1. Boot to the Red Hat Enterprise Linux 6.5 installation disc.
2. Select Install or upgrade an existing system and press Enter.
3. Skip the installation media integrity check and enter the installation process.
4. At the RHEL6 splash screen click Next.
5. Select your language and click Next.
6. Select your keyboard layout and click Next.
7. Select Basic Storage Devices and click Next.
8. Click Yes, discard any data.
9. Give the hypervisor host a hostname.
10. Click Configure Network, and select the NIC to be used for testing. Assign the NIC a static IPv4 address and subnet mask. Click OK.
11. Click Next.
12. Select the time zone to be used and click Next.
13. Provide a password for the root user and click Next.
14. Select Use All Space and click Next.
15. Click Write changes to disk.
16. Click Next.
17. When the OS has finished installing click Reboot.

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 SELinux and the firewall:

```
setenforce 0
sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```

2. Modify `/etc/hosts` to include the IP addresses and the hostnames of the local and client VM.
3. Install 12*c* RPM packages, install package dependencies, and modify kernel parameters:

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

4. Install the following prerequisite package for Oracle ASM support:

```
yum install oracleasm-support
```

5. Create the Oracle groups, assign them to the Oracle user, and set the Oracle user’s 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
```

6. Create the standard Oracle directory structure:

```
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
```

7. Modify the bash profiles for the Oracle and grid environments. Append the following three sets of bash commands to the files /home/oracle/.bash_profile, /home/oracle/data_env and /home/oracle/grid_env:

```
cat >> / home/oracle/.bash_profile << _EOF_1
# Oracle settings for the database environment
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_BASE=/u01/app/oracle
export GRID_HOME=$ORACLE_BASE/product/12.1.0/grid
export DATA_HOME=$ORACLE_BASE/product/12.1.0/dbhome_1
export ORACLE_HOME=$DATA_HOME
export ORACLE_SID=orcl
export BASE_PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
# Two shortcuts for switching between database and grid environments.
alias grid_env='. /home/oracle/grid_env'
alias data_env='. /home/oracle/data_env'
_EOF_1

cat > / home/oracle/data_env << _EOF_2
# switch to Oracle database environment
export ORACLE_SID=orcl
export ORACLE_HOME=$DATA_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
_EOF_2

cat > / home/oracle/grid_env << _EOF_3
# switch to Oracle grid environment
export ORACLE_SID=+ASM
export ORACLE_HOME=$GRID_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
```

```
_EOF_3
```

8. Create disk partitions for ASM using parted:

```
for disk in b h; do
    parted /dev/vd${disk} mklabel gpt
    parted /dev/vd${disk} mkpart primary "1 -1"
done
```

9. Configure ASM and load its modules:

```
oracleasm configure -e -s y -u oracle -g asmadmin
oracleasm init
```

10. Add physical LUNs as ASM disks (only the R930 will have data5 and data6):

```
oracleleasm createdisk logs  /dev/vdb
oracleleasm createdisk data1 /dev/vdc
oracleleasm createdisk data2 /dev/vdd
oracleleasm createdisk data3 /dev/vde
oracleleasm createdisk data4 /dev/vdf
oracleleasm createdisk data5 /dev/vdg
oracleleasm createdisk data6 /dev/vdh
```

Installing Oracle Grid Infrastructure 12*c*

1. Log in as the Oracle user.
2. Unzip linuxamd64_12c_grid_1of2.zip and linuxamd64_12c_grid_2of2.zip.
3. Open a terminal to the unzipped database directory.
4. Set the Oracle grid environment.
5. To start the installer, type `./runInstaller`
6. At the software Updates screen, select Skip updates.
7. In the Select Installation Option screen, select Install and Configure Grid Infrastructure for a Standalone Server, and click Next.
8. Choose the language, and click Next.
9. In the Create ASM Disk Group screen, choose the Disk Group Name, and change redundancy to External.
10. Select the four or six ASM disks that you are planning to use for the database, and click Next.
11. In the Specify ASM Password screen, write the passwords for the ASM users, and click Next.
12. Leave the default Operating System Groups, and click Next.
13. Leave the default installation, and click Next.
14. Leave the default inventory location, and click Next.
15. Under Root script execution, select Automatically run configuration scripts and enter root credentials.
16. In the Prerequisite Checks screen, make sure that there are no errors.
17. In the Summary screen, verify that everything is correct, and click Finish to install Oracle Grid Infrastructure.
18. At one point during the installation, the installation prompts you to execute two configuration scripts as root. Follow the instructions to run the scripts.
19. At the Finish screen, click Close.

20. In preparation for the database installation, create an ASM disk-group, named REDO, for the database redo logs:
    ```
    asmca -silent -createDiskGroup -sysAsmPassword Password1 \
        -asmsnmpPassword Password1 -diskGroupName REDO -disk \
        '/dev/oracleasm/disks/logs' -redundancy EXTERNAL
    ```

Installing Oracle Database 12*c*

1. Log in as the Oracle user.
2. Unzip linuxamd64_12c_database_1_of_2.zip and linuxamd64_12c_database_2_of_2.zip.
3. To start the installer, type `./runInstaller`
4. In the Configure Security Updates screen, uncheck I wish to receive security updates via My Oracle Support, and click Next.
5. If you receive a warning about the email address not being specified, click Yes to continue.
6. At the Software Updates screen, select Skip Software Updates and click Next.
7. In the Select Installation Option screen, select Install database software only, and click Next.
8. Select Single instance database installation, and click Next.
9. Select the language, and click Next.
10. In the Select Database Edition screen, select Enterprise Edition, and click Next.
11. In the Specify Installation Location screen, accept the defaults, and click Next.
12. At the Operating System Groups screen, select the appropriate groups, and click Next.
13. At the summary screen, click Install.

Installing and configuring the Oracle database

1. Log in as the Oracle user.
2. Enter `dbca` and press Enter to open the Database configuration assistant.
3. At the Database Operation screen, select Create Database, and click Next.
4. Under Creation Mode, select Advanced Mode, and click Next.
5. At the Select Template screen, select General Purpose or Transaction Processing. Click Next.
6. Enter a Global database name and the appropriate SID.
7. At the Management Options screen, select Configure Enterprise Manager (EM) Database Express. Click Next.
8. At the Database Credentials screen, select Use the Same Administrative Password for All Accounts. Enter a password, and click Next.
9. At the Network Configuration screen, click Next.
10. At the Storage Configuration screen, select Automatic Storage Management, select +DATA as the Database File Location, and select /backup as the Fast Recovery Area with a 37GB Fast Recover Area size.
11. At the Database Options screen, click Next.
12. At the Initialization Parameters screen, click Next.
13. At the Creation Options, select Create Database, and click Next.
14. At the summary screen, click Finish.
15. Close the Database Configuration Assistant.

    We then configured our database with a custom spfile:

```
orcl.__data_transfer_cache_size=0
orcl.__java_pool_size=33554432
orcl.__large_pool_size=83886080
orcl.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
```

```
orcl.__pga_aggregate_target=1677721600
orcl.__sga_target=5033164800
orcl.__shared_io_pool_size=0
sga_target=12G
orcl.__streams_pool_size=0
*.audit_file_dest='/u01/app/oracle/admin/orcl/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='+DATA/ORCL/CONTROLFILE/current.262.878194059','+DATA/ORCL/CONTRO
LFILE/current.261.878194059'#Restore Controlfile
*.db_block_size=8192
*.db_cache_size=10G
*.db_create_file_dest='+DATA'
*.db_domain=''
*.db_name='orcl'
*.db_recovery_file_dest_size=37580963840
*.db_recovery_file_dest='/backup'
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=orclXDB)'
*.fast_start_mttr_target=180
*.filesystemio_options='setall'
*.log_checkpoints_to_alert=TRUE
*.open_cursors=2000
*.processes=1000
*.query_rewrite_enabled='false'
*.recyclebin='off'
*.remote_login_passwordfile='EXCLUSIVE'
*.sessions=1500
*.statistics_level='typical'
*.timed_statistics=TRUE
*.trace_enabled=FALSE
*.undo_retention=1
*.undo_tablespace='UNDOTBS1'
*.use_large_pages='true'
```

Creating redo logs

1. Log into sqlplus and create two log groups with 19GB files on +LOG:

   ```
   sqlplus / as sysdba
   ALTER DATABASE ADD LOGFILE GROUP 1 ('+LOG/ORCL/ONLINELOG/log1.rdo’)size 19g;
   ALTER DATABASE ADD LOGFILE GROUP 2 ('+LOG/ORCL/ONLINELOG/log2.rdo’)size 19g;
   ```

2. Remove the three default redo log groups.

Configuring and cloning the VMs

Tuning the VMs

We made a number of adjustments to the base VM to enhance performance and make cloning easier.

1. Install the following packages on the server:

```
yum install -y time tuned acpid wget vim nfs-utils openssh-clients man unzip
sysstat numactl
```

2. Run the following commands to allow for easier NIC configuration after cloning:

```
ln -s /dev/null /etc/udev/rules.d/75-persistent-net-generator.rules
rm -f /etc/udev/rules.d/70-persistent-net.rules
```

3. Edit /etc/sysconfig/network-scripts/ifcfg-eth0 to be the following:

```
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
IPADDR=<VM IP ADDRESS>
PREFIX=24
```

4. Edit /etc/fstab:

```
CHANGE:
/dev/mapper/vg_test1-lv_root /                        ext4    defaults        1 1
/dev/mapper/vg_test1-lv_home /home                    ext4    defaults        1 2

TO:
/dev/mapper/vg_test1-lv_root /                        ext4
defaults,nobarrier,noatime,nodiratime        1 1
/dev/mapper/vg_test1-lv_home /home                    ext4
defaults,nobarrier,noatime,nodiratime        1 2
```

5. Run the following commands to sync the time to your NTP server:

```
service ntpd stop
sed -i '/server .*iburst/d' /etc/ntp.conf
echo 'server <NTP SERVER HOSTNAME> iburst prefer' >> /etc/ntp.conf
service ntpd start
chkconfig ntpd on
```

6. Apply the Tuned virtual guest profile:

```
tuned-adm profile virtual-guest
```

Cloning the VMs on the Dell PowerEdge R930:

We used the following script to clone the base VM on the R930:

```
#!/bin/bash
VM_START=$1
VM_END=$2
VM_PREFIX=orcl
VM_SOURCE=orcl01
CONNECT_HOST=r930.test.lan
VM_OSVOL=vg_kvmhost

for VM in `seq $VM_START $VM_END`;
do
  VM_NUM=`printf "%02d" $VM`
  VM_NAME=${VM_PREFIX}${VM_NUM}
#  VM_SOURCE=orcl0`expr \( ${VM} - 1 \) % 4 + 1`
  virt-clone --connect=qemu+ssh://root@${CONNECT_HOST}/system -o $VM_SOURCE -n
$VM_NAME -f /dev/${VM_OSVOL}/${VM_NAME}-os -f /dev/logs/${VM_NAME}-logs -f
/dev/data1/${VM_NAME}-data1 -f /dev/data2/${VM_NAME}-data2 -f
/dev/data3/${VM_NAME}-data3 -f /dev/data4/${VM_NAME}-data4 -f
/dev/data5/${VM_NAME}-data5 -f /dev/data6/${VM_NAME}-data6
done
```

Cloning the VMs on the Dell PowerEdge R910:

We used the following script to clone the base VM on the R910:

```
#!/bin/bash
VM_START=$1
VM_END=$2
VM_PREFIX=orcl
VM_SOURCE=orcl01
CONNECT_HOST=r910.test.lan
VM_OSVOL=vg_legacyr910

for VM in `seq $VM_START $VM_END`;
do
  VM_NUM=`printf "%02d" $VM`
  VM_NAME=${VM_PREFIX}${VM_NUM}
#  VM_SOURCE=orcl0`expr \( ${VM} - 1 \) % 4 + 1`
  virt-clone --connect=qemu+ssh://root@${CONNECT_HOST}/system -o $VM_SOURCE -n
$VM_NAME -f /dev/${VM_OSVOL}/${VM_NAME}-os -f /dev/logs/${VM_NAME}-logs -f
/dev/data1/${VM_NAME}-data1 -f /dev/data2/${VM_NAME}-data2 -f
/dev/data3/${VM_NAME}-data3 -f /dev/data4/${VM_NAME}-data4
done
```

Setting up the HammerDB client

We used a dual-processor auxiliary server running Red Hat Enterprise Linux 6.5, 64-bit, for the HammerDB client. We installed the operating system, following the same instructions as we used for the database VMs. We then installed the HammerDB client software.

Installing HammerDB

We downloaded HammerDB 2.16 64-bit from the following location: hammerora.sourceforge.net. We installed HammerDB according to the installation guide (hammerora.sourceforge.net/hammerdb_install_guide.pdf), including the installation of one additional software package: libXScrnSaver-devel.

Installing Oracle Client libraries for HammerDB

1. Copy the Oracle 12*c* database client software, linuxamd64_12102_client.zip, to the auxiliary server.
2. Extract the archive's contents to directory/client:
   ```
   mkdir /client
   cd /client
   unzip ~/linuxamd64_12102_client.zip
   chmod -R a+rX /client
   ```
3. Log on to the server as the Oracle user.
4. Launch the Oracle Client Installer:
   ```
   /client/runInstaller
   ```
5. In Select Installation Type, select Administrator (1.8 GB) as the installation type, and click Next.
6. In Software Updates, select Skip software updates, and click Next.
7. In Select Product Languages, select English, and click the right-pointing arrow located between the two selection panels to add English to the Selected languages panel. Click Next.
8. In Specify Installation Location, accept the default locations provided, and click Next.
9. In Create Inventory, accept the defaults, and click Next.
10. In Summary, review the information, and click Install to begin installation.
11. In Install Product, follow the instructions to execute the scripts. Click OK when the scripts have completed.
12. In Finish, click Close to exit the installer.

Configuring the database

We modified the following sections of the HammerDB config.xml to build and test our database workload:

```
<?xml version="1.0" encoding="utf-8"?>
<hammerdb>
    <rdbms>Oracle</rdbms>
    <bm>TPC-C</bm>
<oracle>
    <service>
        <system_user>system</system_user>
        <system_password>Password1</system_password>
        <instance>orcl</instance>
    </service>
    <tpcc>
        <schema>
            <count_ware>1</count_ware>
          <num_threads>1</num_threads>
```

```
            <tpcc_user>tpcc</tpcc_user>
            <tpcc_pass>tpcc</tpcc_pass>
            <tpcc_def_tab>tpcctab</tpcc_def_tab>
            <tpcc_ol_tab>tpcctab</tpcc_ol_tab>
            <tpcc_def_temp>temp</tpcc_def_temp>
            <plsql>0</plsql>
            <directory>
            </directory>
          <partition>false</partition>
          <tpcc_tt_compat>false</tpcc_tt_compat>
        </schema>
        <driver>
            <total_iterations>40000000</total_iterations>
            <raiseerror>false</raiseerror>
            <keyandthink>false</keyandthink>
          <checkpoint>false</checkpoint>
            <oradriver>awr</oradriver>
            <rampup>10</rampup>
            <duration>20</duration>
        </driver>
    </tpcc>
```

[UNMODIFIED ORACLE TPC-H SECTIONS OMITTED FOR CLARITY]

```
    <transaction_counter>
        <connect_string>system/Password1@orcl</connect_string>
        <refresh_rate>10</refresh_rate>
        <rac>0</rac>
        <autorange>0</autorange>
    </transaction_counter>
</oracle>
```

[UNMODIFIED MSSQLSERVER, MYSQL, POSTGRESQL, AND REDIS SECTIONS OMITTED FOR CLARITY]

```
  <virtual_user_options>
        <virtual_users>101</virtual_users>
        <user_delay>0</user_delay>
        <repeat_delay>0</repeat_delay>
        <iterations>1</iterations>
        <show_output>1</show_output>
        <log_to_temp>1</log_to_temp>
     <unique_log_name>0</unique_log_name>
     <no_log_buffer>0</no_log_buffer>
    </virtual_user_options>
    <mode>
     <autopilot_options>
          <apmode>disabled</apmode>
          <autopilot_duration>30</autopilot_duration>
          <autopilot_sequence>1000</autopilot_sequence>
     </autopilot_options>
        <slave_options>
            <hostname>localhost</hostname>
            <id>0</id>
        </slave_options>
    </mode>
  <code_highlight>
     <highlight>true</highlight>
  </code_highlight>
</hammerdb>
```

Running HammerDB

The config file above declares 101 virtual users and zero millisecond user/repeat delay. We used a 10 minute ramp-up time and a 20 minute measurement time. We used rman to back up the database and restore between runs. After restoring the database, we rebooted all the VMs as well as the host and allowed the system to reach an idle state before beginning the workload. We ran the test three times and report the median score.

ABOUT PRINCIPLED TECHNOLOGIES

Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
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.

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.