MAXIMIZING ORACLE DATABASE PERFORMANCE WITH INTEL SSD DC SERIES P3600 NVME SSDS ON THE DELL POWEREDGE R930

For critical database workloads that require a high volume of transactions, administrators must seek every advantage to yield maximum performance from each server. Increased performance can mean using fewer servers, potentially saving in space, hardware, and even infrastructure costs. Even workhorse servers like the Intel Xeon processor-powered Dell PowerEdge R930 can accomplish significantly more work when you make strategic upgrades to internal hardware, including the types of drives you use.

In our hands-on tests, Principled Technologies found that configuring the Dell PowerEdge R930 with Intel SSD DC P3600 Series NVMe SSDs could give Oracle® Database 12*c* performance a large boost, delivering up to 2.17 times the new orders per minute (NOPM) compared to SATA SSDs. By making the extra investment in hardware upgrades at the time of purchase (with only an 18 percent higher purchase price), administrators can be confident that they are using all the resources at their disposal to optimize critical transactional database performance, potentially reducing the costs associated with additional servers and corresponding software licenses. Better performance from each server also helps ensure that your business runs smoothly by allowing customers and employees to complete their requests more quickly.

CRITICAL DATABASES DESERVE A BOOST

Every enterprise has databases that are vital to its continued operation. Some organizations may rely on databases to serve customers in an online store, others may have databases full of medical or school records that clients need to access. For other organizations, internal databases are the main driver for employee tasks. No matter the reason, your databases are the backbone of your organization, and anything you can do to help them run better for customers and employees can improve user experience and productivity.

Intel NVMe SSDs are designed to deliver increased IOPS and lower latency than legacy storage standards such as SATA and SAS SSDs, which are themselves a step forward from traditional HDDs. When used in conjunction with the Dell PowerEdge R930, we found that Intel NVMe SSDs could improve database performance significantly for Oracle Database for relatively low additional cost.

To test this, we used a Dell PowerEdge R930 with either eight Intel SATA SSDs or eight Intel SSD DC P3600 Series NVMe SSDs. We configured a 300GB database on six SSDs configured into three RAID 1 and logs on a two disk RAID 1. We ran HammerDB benchmark against the database with no think time to simulate a peak performance scenario. We used a database larger than the amount of system memory to ensure the load would stay on the SSDs. We used the HammerDB NOPM output to compare the performance between the drives. While HammerDB reports TPM and NOPM statistics, NOPM is the preferred metric because it is a value that can be compared across different database systems.

For more about the components we tested, including the benchmark we used, see Appendix A. To learn more about our system configuration, see Appendix B. See Appendix C for our test methodology.

WHAT WE FOUND

While the Dell PowerEdge R930 configuration with SATA SSDs delivered strong database performance, the same configuration with Intel NVMe SSDs provided significantly better performance. As Figure 1 shows, configuring the Dell PowerEdge R930 with Intel NVMe SSDs delivered over twice the Oracle Database performance running an online transaction processing (OLTP) database workload than the same server with SATA SSDs. (Note: Oracle licensing agreements prohibit publishing benchmark scores. Thus, we report only relative performance.)

Figure 1: The Dell PowerEdge R930 with Intel NVMe SSDs outperformed the same server with SATA SSDs on Oracle Database.
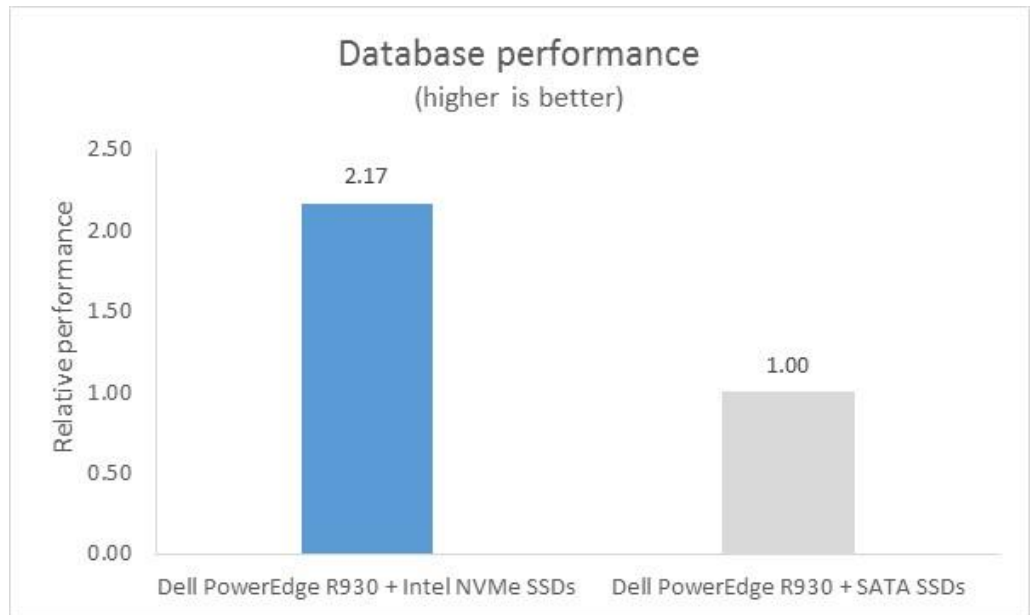
For users, this can mean quicker access to database results because the server can process more orders. When a fast database lets customers search or complete purchases more quickly, they may be more likely to stay on your site, continue browsing, or even return for additional purchases. Internal users, such as employees who rely on database information when providing customer service, can get more done when the server can handle additional requests more quickly.

For administrators, an upgrade to Intel NVMe SSDs can help them maximize the infrastructure they have to meet performance targets that drive the organization. In addition to saving space, administrators benefit from the potential to have less hardware to manage and maintain by consolidating workloads onto fewer servers with more powerful internal hardware.

For businesses, maximizing database performance within each server can mean big savings by avoiding additional server and software license purchases and in datacenter space. We found that the cost of purchasing a new server with Intel SSD DC P3600 Series NVMe SSDs was approximately 18 percent higher than purchasing the same server with a mix of hard drives and SATA SSDs (see Appendix B for configurations). With the additional cost of the drive upgrade, only 18 percent more for hardware, you can double the performance of your Dell PowerEdge R930. Figure 2 details the prices Dell quoted PT for both configurations, as of 5/9/16. These prices do not include tax or shipping.

System	Price (USD)
Dell PowerEdge R930 + Intel NVMe SSDs	$97,150
Dell PowerEdge R930 + SATA SSDs	$82,300

Figure 2: Prices for servers with different drive types, quoted on 5/9/2016.

Ultimately, maximizing server performance can help you utilize resources most efficiently to make sure your organization gets the most out of your datacenter.

CONCLUSION

If your organization runs critical, high-demand databases in environments such as Oracle Database, strong performance is not an option: it's a must-have. Additionally, getting that necessary strong performance out of a single server can be essential for running a space and cost-efficient datacenter. In the Principled Technologies labs, we found that the Dell PowerEdge R930 offered strong performance for such transactional databases when configured with SATA SSDs. When we upgraded the servers to Intel SSD DC P3600 Series NVMe SSDs, performance doubled, increasing by 2.17 times, or 117 percent. If your datacenter needs a new powerhouse server, purchasing your Dell PowerEdge R930 with Intel NVMe SSDs for a cost increase of only 18 percent can double the performance you get from each server. This increases what your infrastructure can do within the same amount of space and lets you ultimately save money that would otherwise be spent purchasing additional servers and software.

APPENDIX A – ABOUT THE COMPONENTS

About the Dell PowerEdge R930

The Dell PowerEdge R930 is a 4U, four-socket server that features the latest from the Intel Xeon processor E7 series. It is a versatile system designed to handle demanding workloads such as large-scale virtualization and massive databases— workloads that are becoming increasingly important for the day-to-day operations of enterprise organizations.

Under the hood, the R930 boasts a bevy of high-class specifications that make it a powerful tool for the modern datacenter. These specifications include:

- 96 DIMM slots—The ample number of slots help leverage cost with performance and capacity needs. It accommodates a larger memory footprint which can be configured with smaller-capacity DIMMs for potential cost-efficiency, or can support up to 6 TB for more memory- intensive use cases.
- 24 Drive bays, configurable with a mix of SAS/SATA HDDs and SSDs as well as optional NVMe PCIe® SSDs for expanded storage capabilities. Two chassis configurations—one with 24 HDD/SSD drive bays, and one with 16 HDD/SSD drive bays and 8 PCIe SSD drive bays—allow you to configure the PowerEdge R930 however your business needs.
- Optional NVMe PCIe SSD drives for data that requires the fastest- available I/O throughput.

To learn more about the Dell PowerEdge R930, visit http://www.dell.com/us/business/p/poweredge-r930/pd.

About HammerDB

HammerDB is an open-source benchmark tool that tests the database performance of many leading databases, including Oracle Database, Microsoft® SQL Server®, PostgreSQL®, MySQLTM, and more. The benchmark includes two built-in workloads derived from industry-standard benchmarks: a transactional OLTP workload and a data warehouse workload. For this study, we used the transactional workload. For more information about HammerDB, visit http://hammerora.sourceforge.net

APPENDIX B – SYSTEM CONFIGURATION INFORMATION

Figure 3 provides detailed configuration information for the test systems.

System	Dell PowerEdge R930
Power supplies	
Total number	4
Vendor and model number	Dell 0GDPF3
Wattage of each (W)	1,100
Cooling fans	
Total number	6
Vendor and model number	Nidec® UltraFlo V12C12BS1M3
Dimensions (h × w) of each	5" x 5"
Volts	2.31
General	
Number of processor packages	4
Number of cores per processor	18
Number of hardware threads per core	2
CPU	
Vendor	Intel
Name	Xeon
Model number	E7-8890 v3
Socket type	LGA2001-1
Core frequency (GHz)	2.50
Bus frequency	9.6 GT/s
L1 cache	32 KB + 32 KB (per core)
L2 cache	256 KB (per core)
L3 cache	45 MB
Platform	
Vendor and model number	Dell PowerEdge R930
BIOS name and version	1.0.9
BIOS settings	Default
Memory module(s)	
Total RAM in system (GB)	256
Vendor and model number	Samsung® M386A4G40DM0-CPB
Type	PC4-2133P
Speed (MHz)	2,133
Speed running in the system (MHz)	2,133
Timing/Latency (tCL-tRCD-tRP-tRASmin)	15-15-15-33
Size (GB)	32
Number of RAM module(s)	8
Chip organization	Double-sided
Rank	Quad

System	Dell PowerEdge R930
Operating system	
Name	Red Hat® Enterprise Linux®
Kernel	3.10.0-229.el7.x86_64
Language	English
RAID controller	
Vendor and model number	PERC H730P
Firmware version	25.3.0.0016
Cache size (MB)	2,048
Local storage	
Hard drive (for OS)	
Vendor and model number	Dell AL13SEB900
Number of drives	2
Size (GB)	900
RPM	10,000
Type	SAS
SATA solid-state drive	
Vendor and model number	Intel SSDSC2BB800G6R
Number of drives	8
Size (GB)	800
RPM	N/A
Type	SATA SSD
NVMe PCIe solid-state drive	
Vendor and model number	Intel SSDPE2ME020T4D
Number of drives	8
Size (GB)	2,000
RPM	N/A
Type	NVMe PCIe SSD
Ethernet adapters	
Vendor and model number	Intel Gigabit 4P i350-t
Type	Gigabit Ethernet

Figure 3: Detailed configuration information for the test server.

APPENDIX C – HOW WE TESTED

Configuring Red Hat Enterprise Linux and Oracle Database 12*c*

We installed Red Hat Enterprise Linux on Dell PowerEdge R930 server and configured settings as we specify below. Screen outputs are in grey boxes.

Installing Red Hat Enterprise Linux

We installed Red Hat Enterprise Linux on the R930 server with the default settings, but created a 16GB swap file as Oracle requires it.

Performing initial configuration tasks

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

1. Disable SELINUX:

   ```
   vi /etc/selinux/config
   SELINUX=disabled
   ```

2. Disable the firewall:

   ```
   systemctl disable firewalld
   ```

3. To update the operating system packages, type the following:

   ```
   yum update -y
   ```

4. To install additional packages, type the following commands:

   ```
   yum install -y acpid wget vim nfs-utils openssh-clients man lsscsi unzip
   smartmontools numactl ipmitool OpenIPMI
   ```

5. Reboot the server.

   ```
   reboot
   ```

6. Install additional packages with the following commands:

   ```
   yum install -y \
   binutils \
   compat-libcap1  \
   compat-libstdc++-33  \
   compat-libstdc++-33.i686  \
   gcc  \
   gcc-c++  \
   glibc  \
   glibc.i686  \
   glibc-devel  \
   glibc-devel.i686  \
   ksh  \
   libgcc  \
   ```

```
    libgcc.i686  \
    libstdc++  \
    libstdc++.i686  \
    libstdc++-devel  \
    libstdc++-devel.i686  \
    libaio  \
    libaio.i686  \
    libaio-devel  \
    libaio-devel.i686  \
    libXext  \
    libXext.i686  \
    libXtst  \
    libXtst.i686  \
    libX11  \
    libX11.i686  \
    libXau  \
    libXau.i686  \
    libxcb  \
    libxcb.i686  \
    libXi  \
    libXi.i686  \
    make  \
    sysstat  \
    unixODBC  \
    unixODBC-devel \
    xorg-x11-xauth \
    xorg-x11-utils
```

7. Edit the sysctl file.

```
    vim /etc/sysctl.conf
  fs.file-max = 6815744
  kernel.sem = 250 32000 100 128
  kernel.shmmni = 4096
  kernel.shmall = 1073741824
  kernel.shmmax = 540945166336
  net.core.rmem_default = 262144
  net.core.rmem_max = 4194304
  net.core.wmem_default = 262144
  net.core.wmem_max = 1048576
  fs.aio-max-nr = 1048576
  net.ipv4.ip_local_port_range = 9000 65500
  vm.nr_hugepages = 102400
  vm.hugetlb_shm_group = 54321
```

8. Apply the changes with the following command:

```
sysctl -p
```

9. Edit the security limits configuration.

```
vim /etc/security/limits.conf
oracle   soft     nofile   1024
oracle   hard     nofile   65536
oracle   soft     nproc    2047
oracle   hard     nproc    16384
oracle   soft     stack    10240
oracle   hard     stack    32768
oracle   soft     memlock  536870912
oracle   hard     memlock  536870912
```

10. Add the necessary groups and users.

```
groupadd -g 54321 oinstall
groupadd -g 54322 dba
groupadd -g 54323 oper
useradd -u 54321 -g oinstall -G dba,oper oracle
```

11. Modify the password for the Oracle user.

```
passwd oracle
```

```
Changing password for user oracle.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

12. Edit the hosts file.

```
vim /etc/hosts
127.0.0.1       R930 R930.localhost.localdomain localhost
localhost.localdomain localhost4 localhost4.localdomain4
::1             R930 R930.localhost.localdomain localhost
localhost.localdomain localhost6 localhost6.localdomain6
```

13. Edit the 90-nproc.conf file.

```
vim /etc/security/limits.d/20-nproc.conf
```
Modifying this line:
```
*            soft    nproc    1024
```
To reflect this change:
```
* - nproc 16384
```

14. Edit the profile file to set environment variables.

```
vim /home/oracle/.bash_profile
# Oracle Settings
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=R930.localhost.localdomain
export ORACLE_BASE=/home/oracle/app/oracle
export GRID_HOME=$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'
```

15. Edit the grid_env file, and adjust additional variables:

```
vim /home/oracle/grid_env
export ORACLE_SID=+ASM
export ORACLE_HOME=$GRID_HOME
export PATH=$ORACLE_HOME/bin:$BASE_PATH

export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
```

16. Edit the db_env file, and adjust additional variables:

```
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
```

17. Edit the scsi_id file.

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

Setting up the SATA SSD storage

We used the following steps to configure the SSD storage prior to setting up ASM.

1. Type the following command to get the ID of each driver.

   `/lib/udev/scsi_id -d /dev/sdb` (Replace sdb with your drive name.)

2. Edit the 99-oracle-asmdevices rules file.

   ```
   vim /etc/udev/rules.d/99-oracle-asmdevices.rules
   KERNEL=="sd?1",ENV{DEVTYPE}=="partition",ENV{ID_SERIAL}=="3644a8420053ac0
   001df888bf1a25baf0",SYMLINK+="oracleasm/mirror01", OWNER="oracle",
   GROUP="dba", MODE="0660"
   KERNEL=="sd?1",ENV{DEVTYPE}=="partition",ENV{ID_SERIAL}=="3644a8420053ac0
   001df888ea1cbba3c1", SYMLINK+="oracleasm/mirror02", OWNER="oracle",
   GROUP="dba", MODE="0660"
   KERNEL=="sd?1",ENV{DEVTYPE}=="partition",ENV{ID_SERIAL}=="3644a8420053ac0
   001df889131f2b65b4", SYMLINK+="oracleasm/mirror03", OWNER="oracle",
   GROUP="dba", MODE="0660"
   KERNEL=="sd?1",ENV{DEVTYPE}=="partition",ENV{ID_SERIAL}=="3644a8420053ac0
   001df889362143ded4", SYMLINK+="oracleasm/mirror04", OWNER="oracle",
   GROUP="dba", MODE="0660"
   ```

3. Execute udevadm and start udev.

   ```
   udevadm control --reload-rules
   reboot
   ```

4. List the ASM devices.

   ```
   ls -l /dev/oracleasm/
   ```
   ```
   lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror01 -> ../sdb1
   lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror02 -> ../sdc1
   lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror03 -> ../sdd1
   lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror04 -> ../sde1
   ```

Setting up the NVMe PCIe SSD storage

We used the following steps to configure the PCIe SSD storage prior to setting up ASM.

1. Run the following command to create SCSI names.

```
for i in `seq 0 7`; do /lib/udev/scsi_id --export -d /dev/nvme${i}n1 |
grep ID_SCSI_SERIAL ; done
ID_SCSI_SERIAL=        PHFT5395002D800HGN
ID_SCSI_SERIAL=        PHFT5383000F800HGN
ID_SCSI_SERIAL=        PHFT53830029800HGN
ID_SCSI_SERIAL=        PHFT5395007E800HGN
ID_SCSI_SERIAL=        PHFT53950020800HGN
ID_SCSI_SERIAL=        PHFT5395002G800HGN
ID_SCSI_SERIAL=        PHFT539500CX800HGN
ID_SCSI_SERIAL=        PHFT53830011800HGN
```

2. Edit the 99-oracle-asmdevices rules file.

```
vim /etc/udev/rules.d/99-oracle-asmdevices.rules
KERNEL=="nvme?n?",ENV{ID_SCSI_SERIAL}!="?*",IMPORT{program}="/lib/udev/sc
si_id -- export --whitelisted -d $tempnode", ENV{ID_BUS}="scsi"
KERNEL=="nvme?n?p1",ENV{DEVTYPE}=="partition",ENV{ID_SCSI_SERIAL}=="PHFT5
395002D800HGN", SYMLINK+="oracleasm/ssd0", OWNER="oracle", GROUP="dba",
MODE="0660"
KERNEL=="nvme?n?p1",ENV{DEVTYPE}=="partition",ENV{ID_SCSI_SERIAL}=="PHFT5
383000F800HGN", SYMLINK+="oracleasm/ssd1", OWNER="oracle", GROUP="dba",
MODE="0660"
KERNEL=="nvme?n?p1",ENV{DEVTYPE}=="partition",ENV{ID_SCSI_SERIAL}=="PHFT5
3830029800HGN", SYMLINK+="oracleasm/ssd2", OWNER="oracle", GROUP="dba",
MODE="0660"
KERNEL=="nvme?n?p1",ENV{DEVTYPE}=="partition",ENV{ID_SCSI_SERIAL}=="PHFT5
395007E800HGN", SYMLINK+="oracleasm/ssd3", OWNER="oracle", GROUP="dba",
MODE="0660"
KERNEL=="nvme?n?p1",ENV{DEVTYPE}=="partition",ENV{ID_SCSI_SERIAL}=="PHFT5
3950020800HGN", SYMLINK+="oracleasm/ssd4", OWNER="oracle", GROUP="dba",
MODE="0660"
KERNEL=="nvme?n?p1",ENV{DEVTYPE}=="partition",ENV{ID_SCSI_SERIAL}=="PHFT5
395002G800HGN", SYMLINK+="oracleasm/ssd5", OWNER="oracle", GROUP="dba",
MODE="0660"
KERNEL=="nvme?n?p1",ENV{DEVTYPE}=="partition",ENV{ID_SCSI_SERIAL}=="PHFT5
39500CX800HGN", SYMLINK+="oracleasm/ssd6", OWNER="oracle", GROUP="dba",
MODE="0660"
KERNEL=="nvme?n?p1",ENV{DEVTYPE}=="partition",ENV{ID_SCSI_SERIAL}=="PHFT5
3830011800HGN", SYMLINK+="oracleasm/ssd7", OWNER="oracle", GROUP="dba",
MODE="0660"
```

3. Execute udevadm and start udev.

```
udevadm control --reload-rules
reboot
```

4. List the ASM devices.

```
ls -l /dev/oracleasm/
```

```
lrwxrwxrwx 1 root root 12 Feb 19 16:46 ssd0 -> ../nvme0n1p1
lrwxrwxrwx 1 root root 12 Feb 19 16:46 ssd1 -> ../nvme1n1p1
lrwxrwxrwx 1 root root 12 Feb 19 16:46 ssd2 -> ../nvme2n1p1
lrwxrwxrwx 1 root root 12 Feb 19 16:46 ssd3 -> ../nvme3n1p1
lrwxrwxrwx 1 root root 12 Feb 19 16:46 ssd4 -> ../nvme4n1p1
lrwxrwxrwx 1 root root 12 Feb 19 16:46 ssd5 -> ../nvme5n1p1
lrwxrwxrwx 1 root root 12 Feb 19 16:46 ssd6 -> ../nvme6n1p1
lrwxrwxrwx 1 root root 12 Feb 19 16:46 ssd7 -> ../nvme7n1p1
```

Installing Oracle Grid Infrastructure for Standalone Server 12*c*

In all test configurations, we used Oracle's recommended approach to managing storage, Automatic Storage Management (ASM). On each configuration, we configured the underlying storage for redundancy, as would be required in nearly all environments. Oracle ASM provides three redundancy levels: Normal for two-way mirroring, High for three-way mirroring, and External, which provides no mirroring but uses redundancy via hardware RAID controllers. On the SATA SSD configuration, which had a Dell PowerEdge RAID Controller (PERC) H730P, we used RAID 1 disk groups, presented those to Oracle ASM, and used External redundancy. On the NVMe PCIe SSD configuration, the SSD devices used no RAID controller, so we used Oracle ASM Normal redundancy for two-way mirroring.

Prior to starting the steps below, we downloaded the Oracle 12 Grid installation and extracted it to the /home/grid directory.

1. Run the GUI installer for Oracle Database using the following commands:

```
ssh –Y oracle@R930_IP_address
grid_env
cd /home/grid
./runInstaller
```

2. Launch the Oracle Grid Infrastructure installation wizard.

3. In Software Updates, select Skip software updates, and click Next.

4. In Installation Options, select Install and Configure Oracle Grid Infrastructure for a Standalone Server, and click Next.

5. In Product Languages, select English and click the right-pointing arrow between the two selection panels to add English to the Selected languages panel. Click Next.

6. In Create ASM Disk Group, click Change Discovery Path.

7. Enter `/dev/oracleasm/*` for the Disk Discovery Path, and click OK. (We left the default path for SAS configuration.)

8. Check the boxes for six drives, and Click Next. (We did not select the last two disks on configuration. We configured the last two drives for logs.) Select Normal redundancy.

9. In ASM Password, select Use same passwords for these accounts. Enter and confirm the password, and click Next.

10. In Operating System Groups, set all Groups to dba. Click Next.

11. To confirm the notifications and continue, click Yes.

12. In Installation Location, accept the default locations provided, and click Next.

13. In Create Inventory, accept the defaults, and click Next.

14. In Root Script Execution, check the box for Automatically run configuration scripts.

15. Select Use "root" user credential, and provide the root password. Click Next.

16. In Summary, review the information, and click Install.

17. To confirm using the privileged user for the installer, click Yes.

18. In Finish, click Close to exit the installer.

Configuring disks for log files

We used the following steps to configure the log volume inside ASM.

1. Start the ASM configuration assistant, type `asmca`.

2. On the Disk Group tab, click Create.

3. On the Create Disk Group pop-up screen, enter LOGS for the Disk Group Name.

4. Select External (None) for Redundancy.

5. Select the disk showing in Disk Path, and click Ok.

6. Click Exit to close the ASM configuration assistant.

Installing Oracle Database 12*c*

Before completing the following steps, we downloaded the Oracle Database 12*c* installation and extracted it to the /home/database directory.

1. Run the GUI installer for Oracle Database using the following commands:
    ```
    ssh –Y oracle@R930_IP_address
    db_env
    cd /home/database
    ./runInstaller
    ```

2. Launch the Oracle Database 12*c* Release 1 Installer.

3. In Configure Security Updates, uncheck the I wish to receive security updates via My Oracle Support checkbox. Click Next.

4. Click Yes to confirm no email provided, and continue.

5. In Software Updates, select Skip software updates, and click Next.

6. In Installation Options, select Install database software only, and click Next.

7. In Grid Installation Options, select Single instance database installation, and click Next.

8. In 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.

9. In Database Edition, select Enterprise Edition, and click Next.

10. In Installation Location, accept the default locations provided, and click Next.

11. In Operating System Groups, accept the defaults, and click Next.

12. In Summary, review the information, and click Install to begin installation.

13. When prompted, follow the instructions to execute the scripts. Click OK when the scripts have completed.

14. In Finish, click Close to exit the installer.

15. When prompted in the GUI installer, run the root shell script to finish the Oracle Database installation.

```
/home/oracle/app/oracle/product/12.1.0/dbhome_1/root.sh
```

Creating the Oracle Database (using DBCA)

1. Launch the Database Configuration Assistant (DBCA).

2. In Database Operations, select Create Database, and click Next.

3. In Creation Mode, select Advanced Mode, and click Next.

4. In Database Template, select the Template for General Purpose or Transaction Processing, and click Next.

5. In Database Identification, type `orcl` for the Global Database Name.

6. Type `orcl` for the SID. Click Next.

7. In Management Options, select Configure Enterprise Manager (EM) Database Express. Click Next.

8. In Database Credentials, select Use the Same Administrative Password for All Accounts.

9. Enter and confirm the administrative password, and click Next.

10. In Network Configuration, check the boxes for all listeners, and click Next.

11. In Storage Locations, select User Common Location for All Database Files. Type `+DATA` into the Database Files Location field.

12. Select Specify Fast Recovery Area. Type `(ORACLE_BASE)/fast_recovery_area` in the Fast Recovery Area field.

13. Set the Fast Recovery Area size to 700 GB, and click Next.

14. In Database Options, accept the defaults, and click Next.

15. In Initialization Parameters and under typical settings, set the Memory Size to 80%, and click next.

16. In Creation Options, select Create Database. Click Customize Storage Locations.

17. In the Customize Storage panel and under Redo Log Groups, select 1.

18. Set the file size to 51,200 MB. Click Apply.

19. Under Redo Log Groups, Select 2.

20. Set the file size to 51,200 MB. Click Apply.

21. Under Redo Log Groups, Select 3.

22. Set the file size to 51,200 MB. Click Apply.

23. To exit the Customize Storage panel, click Ok.

24. Click Next.

25. Review the Summary. To complete the database creation, click Finish.

26. Review the information on the screen, and click Exit.

27. To exit the DBCA, click Close.

Configuring Oracle Tablespaces and redo log

Alter the tablespaces on both systems as shown below. Type `sqlplus / as sysdba` to enter SQL prompt.

```
ALTER DATABASE ADD LOGFILE GROUP 11 ( '/tmp/temp1.log' ) SIZE 50M;
ALTER DATABASE ADD LOGFILE GROUP 12 ( '/tmp/temp2.log' ) SIZE 50M;

ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM CHECKPOINT;

ALTER DATABASE DROP LOGFILE GROUP 1;
ALTER DATABASE DROP LOGFILE GROUP 2;
ALTER DATABASE DROP LOGFILE GROUP 3;

ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM CHECKPOINT;

ALTER DATABASE DROP LOGFILE GROUP 1;
ALTER DATABASE DROP LOGFILE GROUP 2;
ALTER DATABASE DROP LOGFILE GROUP 3;

-- DELETE OLD REDO LOG FILES IN ASM MANUALLY USING ASMCMD HERE --

alter system set "_disk_sector_size_override"=TRUE scope=both;

ALTER DATABASE ADD LOGFILE GROUP 1 ( '+LOGS/orcl/redo01.log' ) SIZE 50G
BLOCKSIZE 4k;
ALTER DATABASE ADD LOGFILE GROUP 2 ( '+LOGS/orcl/redo02.log' ) SIZE 50G
BLOCKSIZE 4k;
ALTER DATABASE ADD LOGFILE GROUP 3 ( '+LOGS/orcl/redo03.log' ) SIZE 50G
BLOCKSIZE 4k;

ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM CHECKPOINT;

ALTER DATABASE DROP LOGFILE GROUP 11;
ALTER DATABASE DROP LOGFILE GROUP 12;
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM CHECKPOINT;

ALTER DATABASE DROP LOGFILE GROUP 11;
ALTER DATABASE DROP LOGFILE GROUP 12;

HOST rm -f /tmp/temp*.log
```

```
CREATE BIGFILE TABLESPACE "TPCC"
 DATAFILE '+DATA/orcl/tpcc.dbf' SIZE 800G AUTOEXTEND ON NEXT 1G
 BLOCKSIZE 8K
 EXTENT MANAGEMENT LOCAL AUTOALLOCATE
 SEGMENT SPACE MANAGEMENT AUTO;

CREATE BIGFILE TABLESPACE "TPCC_OL"
 DATAFILE '+DATA/orcl/tpcc_ol.dbf' SIZE 400G AUTOEXTEND ON NEXT 1G
 BLOCKSIZE 16K
 EXTENT MANAGEMENT LOCAL AUTOALLOCATE
 SEGMENT SPACE MANAGEMENT AUTO;

ALTER DATABASE DATAFILE '+DATA/orcl/undotbs01.dbf' RESIZE 32760M;
```

Configuring the Oracle pfile

Alter the Oracle pfile as shown below. Then to make Oracle use it, enter the following, and restart oracle:

```
 CREATE SPFILE = '+DATA/orcl/spfileorcl.ora' FROM PFILE =
'/home/oracle/app/oracle/product/12.1.0/dbhome_1/pfile.ora';


orcl.__oracle_base='/home/oracle/app/oracle'#ORACLE_BASE set from
environment
_disk_sector_size_override=TRUE
_enable_NUMA_support=TRUE
_kgl_hot_object_copies=4
_shared_io_pool_size=512m
aq_tm_processes=0
audit_file_dest='/home/oracle/app/oracle/admin/orcl/adump'
audit_trail='NONE'
compatible='12.1.0.2.0'
control_files='+DATA/orcl/control01.ctl','+DATA/orcl/control02.ctl'
db_16k_cache_size=32g
db_block_size=8192
db_cache_size=128g
db_create_file_dest='+DATA'
db_domain=''
db_name='orcl'
db_recovery_file_dest_size=700g
db_recovery_file_dest='/home/oracle/app/oracle/fast_recovery_area'
db_writer_processes=4
diagnostic_dest='/home/oracle/app/oracle'
disk_asynch_io=TRUE
dispatchers='(PROTOCOL=TCP) (SERVICE=orclXDB)'
dml_locks=500
fast_start_mttr_target=180
java_pool_size=4g
job_queue_processes=0
large_pool_size=4g
local_listener='LISTENER_ORCL'
lock_sga=TRUE
log_buffer=402653184
log_checkpoint_interval=0
```

```
log_checkpoint_timeout=0
log_checkpoints_to_alert=TRUE
open_cursors=2000
parallel_max_servers=0
parallel_min_servers=0
pga_aggregate_target=5g
plsql_code_type='NATIVE'
plsql_optimize_level=3
processes=1000
recovery_parallelism=30
remote_login_passwordfile='EXCLUSIVE'
replication_dependency_tracking=FALSE
result_cache_max_size=0
sessions=1500
shared_pool_size=9g
statistics_level='BASIC'
timed_statistics=FALSE
trace_enabled=FALSE
transactions=2000
transactions_per_rollback_segment=1
undo_management='AUTO'
undo_retention=1
undo_tablespace='UNDOTBS1'
use_large_pages='ONLY'
```

Setting up the HammerDB client

We used a dual-processor server running Red Hat Enterprise Linux 7 for the HammerDB client. We followed the installation steps at the beginning of this appendix to install Red Hat Enterprise Linux, but installed the GUI. We then installed the HammerDB client software.

Installing HammerDB

Download and install version 2.19 on the Red Hat client. We downloaded HammerDB from the following location: hammerora.sourceforge.net/download.html. We installed HammerDB according to the installation guide (hammerora.sourceforge.net/hammerdb_install_guide.pdf).

Installing HammerDB Oracle libraries

Complete the following steps on both systems.

1. Launch the Oracle Client Installer.
2. In Select Installation Type, select Administrator (1.8 GB) as the installation type, and click Next.
3. In Software Updates, select Skip software updates, and click Next.
4. 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.
5. In Specify Installation Location, accept the default locations provided, and click Next.
6. In Create Inventory, accept the defaults, and click Next.
7. In Summary, review the information, and click Install to begin installation.
8. In Install Product, follow the instructions to execute the scripts. Click OK when the scripts have completed.
9. In Finish, click Close to exit the installer.

Configuring the database

We used the TPC-C build schema build options for Oracle inside HammerDB to build the database. We set the following options in the build schema.

```
Oracle Service Name = R930_IP_addres/orcl
System user = system
System User Password = Password1
TPC-C User = tpcc
TPC-C User Password = tpcc
TPC-C Default Tablespace = tpcc
Order Line Tablespace = tpcc_ol
TPC-C Temporary Tablespace = temp
TimesTen Database Commatible = unchecked
Partition Order Line Table = checked
Number of Warehouses = 5000
Virtual Users to Build Schema = 60
Use PL/SQL Server Side Load = unchecked
Server Side Log Directory = /tmp
Total Transactions per User = 1000000000
```

Running HammerDB

We ran HammerDB by filling in the appropriate information for the driver options. We tested with a 30-minute ramp up time and 30-minute test duration. We used 101 virtual users with 0-ms user delay and repeat delay. We used rman to back up the database and restore between runs.

ABOUT PRINCIPLED TECHNOLOGIES

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.