DELL POWEREDGE R920 RUNNING ORACLE DATABASE: BENEFITS OF UPGRADING WITH NVME EXPRESS FLASH PCIE SSDS

Performance is paramount for companies running Oracle database applications, which often demand low latencies and extreme IOPS capabilities from the storage subsystem. So, it is important to select a server with not only the latest processor technology and large RAM capacity, but also storage upgrade options that can provide extreme service levels. The new Dell PowerEdge R920, powered by the new Intel® Xeon® processor E7 v2 family, is designed to deliver the performance that mission-critical databases require. By upgrading this server with NVMe Express Flash PCIe SSDs, you can take its strong base performance to an even greater level.

In the Principled Technologies labs, we tested two Dell PowerEdge R920 servers running Oracle Database 12*c* with an OLTP TPC-C-like workload, the first with standard serial attached SCSI (SAS) hard disks, and the second with NVMe Express Flash PCIe SSDs. The upgraded configuration with PCIe SSDs delivered 14.9 times the database performance of the server with standard hard disks. While the base configuration delivered excellent performance, the dramatic performance improvement offered with the NVMe Express Flash PCIe SSDs makes this upgrade a wise investment for any business that seeks to better meet the needs of its Oracle database users.

MARCH 2014
A PRINCIPLED TECHNOLOGIES TEST REPORT
Commissioned by Dell Inc.

WHEN YOU UPGRADE, DATABASE PERFORMANCE SKYROCKETS

The Dell PowerEdge R920, which uses the new Intel Xeon processor E7 v2 family, is Dell's fastest four-socket 4U server. Dell designed it to handle extremely demanding, mission-critical workloads such as enterprise resource planning (ERP), e-commerce, large-scale virtualization, and very large databases. In its standard configuration, it has 24 2.5-inch disk bays and supports up to 24 SAS hard disks. The NVMe Express Flash PCIe SSD configuration of the Dell PowerEdge R920 supports up to eight PCIe SSDs and up to 16 SAS hard disks in the remaining bays. This provides the extreme speed of eight high-performing disks and the reliable storage capacity of 16 SAS disk bays.

While the Dell PowerEdge R920 in its standard configuration supports Oracle database well, we wanted to learn how performance would improve when we used the configuration with NVMe Express Flash PCIe SSDs. To do this, we used a benchmark tool that tests the performance of many leading databases. As Figure 1 shows, the second configuration, using NVMe Express Flash PCIe SSDs, had a huge increase in Oracle database performance—the Dell PowerEdge R920 with PCIe SSDs could perform almost 15 times as much work as the standard configuration with SAS hard disks. Keep in mind that the upgraded configuration achieved this enormous performance advantage while using only one-third the number of disks (eight SSDs vs. the 24 SAS disks in the base configuration).

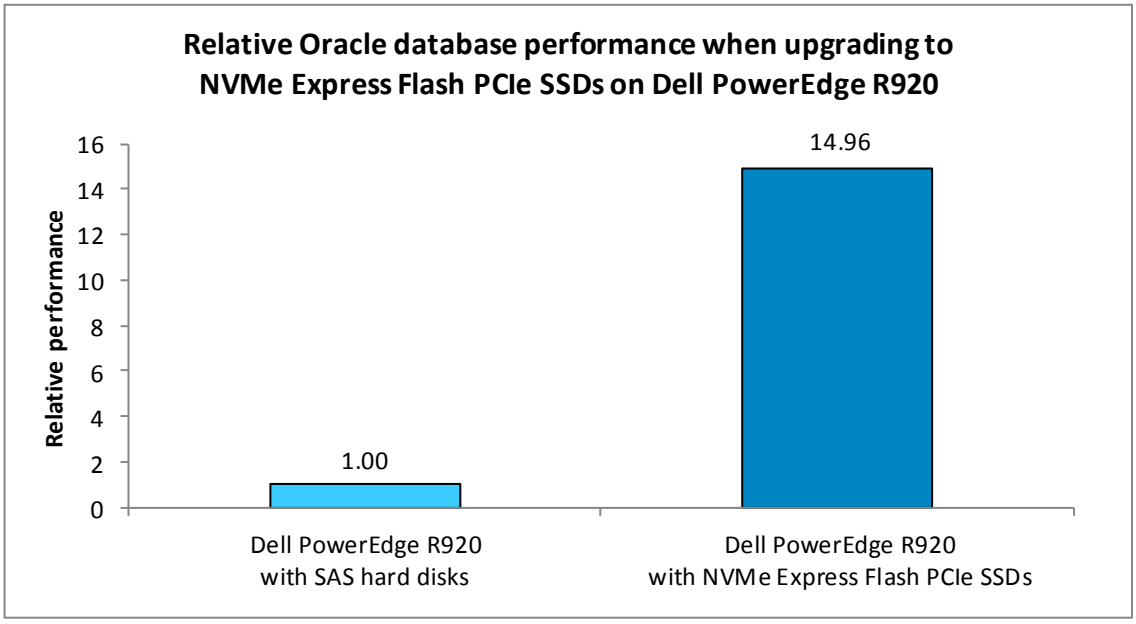

Figure 1: The dramatic performance increase we found between the two server configurations.

We normalized performance to the base configuration because Oracle does not permit publishing specific benchmark results. However, the server, even in its base configuration, delivered very strong results.

Figure 2 shows the performance improvements of our testing with the HammerDB benchmark tool using several different I/O workload profiles. To show performance advantages under several I/O workload profiles, we modified the fast_start_mttr_target parameter in Oracle to use three different settings—60 seconds, 120 seconds, and 180 seconds—and then executed the test each time with the new setting. The fast_start_mttr_target setting allows you to target the number of seconds the database will take to perform crash recovery. The faster the Oracle instance can recover, the quicker the database is back up and available for users.

When fast_start_mttr_target is a smaller target recovery time, say 60 seconds, the database instance must flush modified buffer pages to disk more often during the workload, creating a heavier I/O load on the underlying storage subsystem. When fast_start_mttr_target is a larger target recovery time, for example 180 seconds or more, the less I/O load is incurred on the storage subsystem.

As Figure 2 shows, the Dell PowerEdge R920 with the PCIe SSD configuration handled the increased I/O requirements much better at all three fast_start_mttr_target configuration points. As the fast_start_mttr_target setting was reduced to target a shorter recovery period, the performance advantage of the NVMe Express Flash PCIe SSDs over the SAS disks increased. While the base configuration of the PowerEdge R920 with SAS drives provided solid performance, higher-demand database activity performed better on the NVMe Express Flash PCIe SSD configuration.

fast_start_mttr_target setting	Performance improvement with NVMe Express Flash PCIe SSDs
60 seconds	14.96x
120 seconds	12.93x
180 seconds	6.71x

Figure 2: Relative HammerDB benchmark advantage of the PCIe SSD configuration over the standard configuration

In both test configurations, we used Oracle's recommended approach to managing storage, Automatic Storage Management (ASM). On each server, 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 base 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 PCIe SSD configuration, the SSD devices used no RAID controller, so we used Oracle ASM Normal redundancy for two-way mirroring.[1]

[1] Learn more about ASM mirroring at docs.oracle.com/cd/E11882_01/server.112/e18951/asmcon.htm#OSTMG94058.

The value of faster I/O

The improved storage performance that comes when you upgrade your Oracle database server with NVMe Express Flash PCIe SSDs can benefit your company in a number of ways:

- By improving service level agreements by lowering database response time and/or supporting more users
- By speeding rebuild times in case of a database crash
- By reducing database maintenance time
- By increasing user satisfaction
- By lowering costs as you eliminate underperforming hardware

RAW I/O PERFORMANCE SKYROCKETS, TOO

Unlike database performance, which includes overhead incurred by applications, queries and a software layer, I/O performance is a raw measure of the storage subsystem's ability to process data, and is often many times greater than application-perceived performance.

In addition to database performance, we investigated the raw I/O performance of the NVMe Express Flash PCIe SSDs versus the SAS disks. To measure I/O on the two server configurations, we used the Flexible I/O benchmark tool, also known as Fio. Figure 3 shows our results: In both of the Fio tests we performed, the Dell PowerEdge R920 with the NVMe Express Flash PCIe SSDs dramatically outperformed the base configuration with SAS hard disks.

Note that the random write for SSD configuration of 313,687 IOPS reflects a mirrored configuration approximation. The cumulative raw disk output for the combined devices was twice that, and then we divided that by two to reflect the ASM Normal two-way mirroring configuration we used in database testing.

We ran the Fio tests in a non-mirrored configuration, as the PCIe bus did not have RAID capabilities. We divided the actual results, 627,374 IOPS, by half in order to approximate a RAID1 mirrored write operation. See Appendix B for more details on our configuration.

	Dell PowerEdge R920 with SAS hard disks	Dell PowerEdge R920 with NVMe Express Flash PCIe SSDs	Performance improvement with NVMe Express Flash PCIe SSDs
Read (8k, random, 64 queue)			
Number of disk groups SAS: two disks per disk group SSD: one disk per disk group	10	8	
Total IOPS	8,917	1,719,545	192.8x
Write (8k, random, 64 queue)			
Number of disk groups SAS: two disks per disk group SSD: one disk per disk group	10	8	
Total IOPS	5,904	313,687*	53.1x

Figure 3: Fio benchmark test results. Higher numbers of IOPS are better. *Write IOPs for NVMe totaled 627,374 IOPS, and we halved the result to mathematically approximate a mirrored result.

ABOUT THE COMPONENTS WE TESTED

About the Dell PowerEdge R920

The Dell PowerEdge R920 is the fastest four-socket 4U server from Dell. Designed to provide expanding scalable performance for large enterprises, it supports up to

- 96 DIMMs of memory
- 24 internal disks
- 8 NVMe Express Flash disks (with the optional PCIe backplane)
- 10 PCIe Gen3/Gen2 slots
- 12Gb/s SAS disks

It also offers a 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.

Uses the Intel Xeon processor E7 v2 family

The PowerEdge R920 makes use of the new Intel Xeon processor E7 v2 family, which Intel designed to support mission-critical, high-performance workloads by adding up to 50 percent more cores/threads and 25 percent more cache to provide significant jumps in performance from previous releases. The Intel Xeon processor E7 v2 family provides up to 6TB DDR3 memory, supports up to 24 DDR3 DIMMs per socket, and supports up to 1,600MHz DDR3 speeds to improve performance and increase scalability.

The Intel Xeon processor E7 v2 family supports 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 1, MCA Recovery – Execution Path, MCA I/O, and PCIe Live Error Recovery.

Supports demanding workloads

The PowerEdge R920 can handle extremely demanding, mission-critical workloads such as enterprise resource planning (ERP), e-commerce, large-scale virtualization, and very large databases. It is particularly well suited to the following workloads and environments:

- Accelerating large corporate-wide applications (ERP, CRM, Business Intelligence)
- Implementing very large traditional databases or in-memory databases
- Consolidating enterprise workloads with large-scale virtualization
- Migrating from expensive and outdated RISC hardware to a future-ready data center

To learn more about the Dell PowerEdge R920 server, see www.dell.com/us/business/p/poweredge-r920/pd.

About Dell PowerEdge NVMe Express Flash PCIe SSD

The PowerEdge NVMe Express Flash PCIe SSD is high-performance storage ideal for solutions requiring low latency, high IOPS, and enterprise-class storage reliability and serviceability. A PCIe Gen3-compliant device, the NVMe Express Flash PCIe SSD can be configured as storage cache or as a primary storage device in demanding enterprise environments, such as enterprise blade and rack servers, video-on-demand servers, Web accelerators, and virtualization appliances.

NVM Express is an optimized, high-performance, scalable host controller interface with a streamlined register interface and command set built to handle non-volatile memory (NVM). It is designed to for enterprise, datacenter, and client systems that use PCIe SSDs.

According to the NVMHCI Work Group, a group including more than 90 storage companies, "NVM Express significantly improves both random and sequential performance by reducing latency, enabling high levels of parallelism, and streamlining the command set while providing support for security, end-to-end data protection, and other Client and Enterprise features users need. NVM Express provides a standards-based approach enabling broad ecosystem adoption and PCIe SSD interoperability."

Learn more about the Dell PowerEdge NVMe Express Flash PCIe SSD at www.dell.com/learn/us/en/04/campaigns/poweredge-express-flash. To learn more about the NVM Express interface, visit www.nvmexpress.org/

CONCLUSION

Strong server performance is essential to companies running Oracle Database. The new Dell PowerEdge R920 provides strong performance in its base configuration with 24 SAS hard disks, but this performance gets an enormous boost when running the configuration containing NVMe Express Flash PCIe SSDs. In our testing, the upgraded configuration of the Dell PowerEdge R920 delivered 14.9 times the database performance of the base configuration. In addition, in testing the raw I/O throughput of the NVMe Express Flash PCIe SSDs, we saw as much as 192.8 times the IOPS as compared to the base configuration. Given that the storage subsystem is critical in servers and specifically database applications, the performance improvements offered by NVMe Express Flash PCIe SSDs can lead to great service improvements for your customers, making this upgrade a very wise investment.

APPENDIX A: DETAILED SYSTEM CONFIGURATION

Figure 4 provides detailed configuration information for the test systems.

System	Dell PowerEdge R920 (PCIe SSD configuration)	Dell PowerEdge R920 (SAS configuration)
Power supplies		
Total number	4	4
Vendor and model number	Dell 0GDPF3	Dell 0GDPF3
Wattage of each (W)	1,100	1,100
Cooling fans		
Total number	6	6
Vendor and model number	Nidec® UltraFlow V12C12BS1M3 J87TW-A00	Nidec UltraFlow V12C12BS1M3 J87TW-A00
Dimensions (h x w) of each	5" × 5"	5" × 5"
Volts	12	12
Amps	2.31	2.31
General		
Number of processor packages	4	4
Number of cores per processor	15	15
Number of hardware threads per core	2	2
System power management policy	Max Performance	Max Performance
CPU		
Vendor	Intel	Intel
Name	Xeon	Xeon
Model number	E7-4880 v2	E7-4880 v2
Socket type	LGA 2011	LGA 2011
Core frequency (GHz)	2.5	2.5
Bus frequency	8 GT/s	8 GT/s
L1 cache	32 + 32 KB (per core)	32 + 32 KB (per core)
L2 cache	256 KB (per core)	256 KB (per core)
L3 cache	37.5 MB	37.5 MB
Platform		
Vendor and model number	Dell PowerEdge R920	Dell PowerEdge R920
Motherboard model number	0Y4CNC	0Y4CNC
BIOS name and version	1.0.6	1.0.6
BIOS settings	Defaults	Defaults
Memory module(s)		
Total RAM in system (GB)	256	256
Vendor and model number	Samsung M393B2G70BH0-YK0	Samsung M393B2G70BH0-YK0
Type	PC3L-12800R	PC3L-12800R
Speed (MHz)	1,600	1,600
Speed running in the system (MHz)	1,333	1,333

System	Dell PowerEdge R920 (PCIe SSD configuration)	Dell PowerEdge R920 (SAS configuration)
Timing/Latency (tCL-tRCD-tRP-tRASmin)	11-11-11-35	11-11-11-35
Size (GB)	16	16
Number of RAM module(s)	16	16
Chip organization	Double-sided	Double-sided
Rank	2Rx4	2Rx4
Operating system		
Name	Red Hat® Enterprise Linux®	Red Hat Enterprise Linux
Build number	6.5	6.5
File system	ext4	ext4
Kernel	2.6.32-431.el6.x86_64	2.6.32-431.el6.x86_64
Language	English	English
RAID controller		
Vendor and model number	Dell PERC H730P	Dell PERC H730P
Firmware version	25.2.0.0013	25.2.0.0013
Cache size (GB)	2	2
RAID configuration	2 × RAID1 (OS)	12 × RAID1 pairs
Hard disk types		
Hard disks (OS)		
Vendor and model number	Dell ST9900805SS	Dell ST9900805SS
Number of disks	2	2
Size (GB)	900	900
RPM	10K	10K
Type	SAS	SAS
Hard disks (database and logs)		
Vendor and model number	Dell MZ-WEI4000	Dell ST9300653SS
Number of disks	8	22
Size (GB)	400	300
RPM	N/A	15K
Type	PCIe SSD	SAS
Ethernet adapters		
Vendor and model number	Intel I350 Quad-port 1Gb Daughter card	Intel I350 Quad-port 1Gb Daughter card
Type	PCI-e	PCI-e
USB ports		
Number	4	4
Type	USB 2.0	USB 2.0

Figure 4: Detailed system configuration information.

APPENDIX B: DETAILED TEST METHODOLOGY

About our test tools

HammerDB

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 (TPCH-like) workload. For this study, we used the transactional workload. Our tests were not official TPC results and are not comparable in any manner. For more information about HammerDB, visit hammerora.sourceforge.net.

Flexible I/O (Fio) 2.1.4

Fio is a freely available I/O tool used to stress hardware and reports results in IOPS (input/output operations per second). We downloaded and used Fio version 2.1.4 for testing (pkgs.repoforge.org/fio/fio-2.1.4-1.el6.rf.x86_64.rpm).

Overview of our configuration

We used all eight disks in the SSD configuration for random read and writes and 20 disks in 10 two-disk RAID 1 configurations for the SAS disks. Thus, the SAS configuration consisted of 10 two-disk groups while the SSD configuration had eight single-disk groups. The reason for this configuration is the SSDs do not have a RAID controller for use as the SAS disks do. For the Oracle testing, we presented these configurations to Automatic Storage Management, so we ran Fio in this manner to reflect the Oracle configuration. This simulated the disk configuration holding the database for testing. For Fio, we used an 8k block size to simulate Oracle database configuration.

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

We installed Red Hat Enterprise Linux on both Dell PowerEdge R920 servers 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 Intel server, then configured settings as we specify below.

1. Insert the Red Hat Enterprise 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 your 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. Click Yes, discard any data at the Storage Device Warning.
9. Insert your hostname, and click Next.
10. Select the nearest city in your time zone, and click Next.
11. Enter your root password, and click Next.
12. Select Create Custom Layout, and click Next.
13. Select the install drive and click Create. (Create the following volumes and size: Root = 300GB, Home = 500GB, Boot = 200MB, SWAP = 20GB)
14. Click Next.

15. Click Write changes to disk at the popup window.
16. Select the appropriate Data Store Devices and select where the Bootloader will go, and click Next.
17. Select Software Basic Server, and click Next. Linux installation begins.
18. When the installation completes, select Reboot to restart the server.

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. Set CPU Governor type.
```
vi /etc/sysconfig/cpuspeed
GOVERNOR=performance
```
3. Disable the firewall for IPv4 and IPv6.
```
chkconfig iptables off
chkconfig ip6tables off
```
4. To update the operating system packages, type the following:
```
yum update -y
```
5. To install additional packages, type the following commands:
```
yum install -y acpid cpuspeed wget vim nfs-utils openssh-clients man
lsscsi unzip smartmontools numactl ipmitool OpenIPMI
```
6. Reboot the server.
```
reboot
```
7. 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
```

8. 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 = 4398046511104
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 = 262144
vm.hugetlb_shm_group = 54321
```

9. Apply the changes with the following command:

```
sysctl -p
```

10. 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
```

11. 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
```

12. 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.
```

13. Edit the hosts file.

```
vim /etc/hosts
```

```
127.0.0.1       R920 R920.localhost.localdomain localhost
localhost.localdomain localhost4 localhost4.localdomain4
::1             R920 R920.localhost.localdomain localhost
localhost.localdomain localhost6 localhost6.localdomain6
```

14. Edit the 90-nproc.conf file.

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

Modifying this line:

```
*          soft    nproc    1024
```

To reflect this change:

```
* - nproc 16384
```

15. Edit the profile file to set environment variables.

```
vim /home/oracle/.bash_profile
```

```
# Oracle Settings
export TMP=/tmp
export TMPDIR=$TMP

export ORACLE_HOSTNAME=R920.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'
```

16. 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
```

17. 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
```

18. Edit the scsi_id file.

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

Setting up the SAS storage

We used the steps below to configure the SAS storage prior to setting up ASM.

1. 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}=="36c81f660d8d581001a9a10580658268a",
SYMLINK+="oracleasm/mirror01", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a106c07885c76",
SYMLINK+="oracleasm/mirror02", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a108f09a3aecc",
SYMLINK+="oracleasm/mirror03", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a10ad0b720998",
SYMLINK+="oracleasm/mirror04", OWNER="oracle", GROUP="dba", MODE="0660"
```

```
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a10c00c8d5153",
SYMLINK+="oracleasm/mirror05", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a10d20da90647",
SYMLINK+="oracleasm/mirror06", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a110c1118728c",
SYMLINK+="oracleasm/mirror07", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a111e1229ba5a",
SYMLINK+="oracleasm/mirror08", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a1132135878df",
SYMLINK+="oracleasm/mirror09", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a114814ac573a",
SYMLINK+="oracleasm/mirror10", OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="sd?1", ENV{DEVTYPE}=="partition",
ENV{ID_SERIAL}=="36c81f660d8d581001a9a115c15d5b8ce",
SYMLINK+="oracleasm/mirror11", OWNER="oracle", GROUP="dba", MODE="0660"
```

2. Execute udevadm and start udev.

```
udevadm control --reload-rules
start_udev
```

3. 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
lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror05 -> ../sdf1
lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror06 -> ../sdg1
lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror07 -> ../sdh1
lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror08 -> ../sdi1
lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror09 -> ../sdj1
lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror10 -> ../sdk1
lrwxrwxrwx 1 root root 7 Feb 24 19:17 mirror11 -> ../sdl1
```

Setting up the PCIe SSD storage

We used the steps below 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 scsi_id --export -d /dev/nvme${i}n1 | grep
ID_SCSI_SERIAL ; done
```

```
ID_SCSI_SERIAL=      S1J0NYADC00150
ID_SCSI_SERIAL=      S1J0NYADC00033
ID_SCSI_SERIAL=      S1J0NYADC00111
ID_SCSI_SERIAL=      S1J0NYADC00146
```

```
ID_SCSI_SERIAL=      S1J0NYADC00136
ID_SCSI_SERIAL=      S1J0NYADC00104
ID_SCSI_SERIAL=      S1J0NYADC00076
ID_SCSI_SERIAL=      S1J0NYADC00048
```

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}="scsi_id --
export --whitelisted -d $tempnode", ENV{ID_BUS}="scsi"
KERNEL=="nvme?n?p1", ENV{DEVTYPE}=="partition",
ENV{ID_SCSI_SERIAL}=="S1J0NYADC00150", SYMLINK+="oracleasm/ssd0",
OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="nvme?n?p1", ENV{DEVTYPE}=="partition",
ENV{ID_SCSI_SERIAL}=="S1J0NYADC00033", SYMLINK+="oracleasm/ssd1",
OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="nvme?n?p1", ENV{DEVTYPE}=="partition",
ENV{ID_SCSI_SERIAL}=="S1J0NYADC00111", SYMLINK+="oracleasm/ssd2",
OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="nvme?n?p1", ENV{DEVTYPE}=="partition",
ENV{ID_SCSI_SERIAL}=="S1J0NYADC00146", SYMLINK+="oracleasm/ssd3",
OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="nvme?n?p1", ENV{DEVTYPE}=="partition",
ENV{ID_SCSI_SERIAL}=="S1J0NYADC00136", SYMLINK+="oracleasm/ssd4",
OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="nvme?n?p1", ENV{DEVTYPE}=="partition",
ENV{ID_SCSI_SERIAL}=="S1J0NYADC00104", SYMLINK+="oracleasm/ssd5",
OWNER="oracle", GROUP="dba", MODE="0660"
KERNEL=="nvme?n?p1", ENV{DEVTYPE}=="partition",
ENV{ID_SCSI_SERIAL}=="S1J0NYADC00076", SYMLINK+="oracleasm/ssd6",
OWNER="oracle", GROUP="dba", MODE="0660"

KERNEL=="nvme?n?p1", ENV{DEVTYPE}=="partition",
ENV{ID_SCSI_SERIAL}=="S1J0NYADC00048", SYMLINK+="oracleasm/ssd7",
OWNER="oracle", GROUP="dba", MODE="0660"
```

3. Execute udevadm and start udev.

```
udevadm control --reload-rules
start_udev
```

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*

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@R920_IP_address
db_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/nvme*` for the Disk Discovery Path, and click OK. (We left the default path for SAS configuration.)
8. Check the boxes for all drives, and Click Next. (We did not select the last disk on SAS configuration. We configured the last drive for logs in the SAS configuration.)
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. Click Yes to confirm the notifications and continue.
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 to begin installation.
17. Click Yes to confirm using the privileged user for the installer.
18. In Finish, click Close to exit the installer.

Installing Oracle Database 12*c*

Prior to starting the steps below, 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@R920_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 400 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. Click Remove and when prompted, click Yes.
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.

```
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;

-- BEGIN: SSD REDO LOGS --
ALTER DATABASE ADD LOGFILE GROUP 1 ( '+DATA/orcl/redo01.log' ) SIZE 50G
BLOCKSIZE 4k;
ALTER DATABASE ADD LOGFILE GROUP 2 ( '+DATA/orcl/redo02.log' ) SIZE 50G
BLOCKSIZE 4k;
-- END: SSD REDO LOGS --

-- BEGIN: SAS REDO LOGS --
ALTER DATABASE ADD LOGFILE GROUP 1 ( '+REDO/orcl/redo01.log' ) SIZE 50G;
ALTER DATABASE ADD LOGFILE GROUP 2 ( '+REDO/orcl/redo02.log' ) SIZE 50G;
-- END: SAS REDO LOGS --

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

SET SERVEROUTPUT ON
DECLARE
  lat  INTEGER;
  iops INTEGER;
  mbps INTEGER;
BEGIN
-- DBMS_RESOURCE_MANAGER.CALIBRATE_IO (<DISKS>, <MAX_LATENCY>, iops, mbps,
lat);
   DBMS_RESOURCE_MANAGER.CALIBRATE_IO (2000, 10, iops, mbps, lat);

  DBMS_OUTPUT.PUT_LINE ('max_iops = ' || iops);
  DBMS_OUTPUT.PUT_LINE ('latency  = ' || lat);
  dbms_output.put_line('max_mbps = ' || mbps);
end;
/

CREATE BIGFILE TABLESPACE "TPCC"
 DATAFILE '+DATA/orcl/tpcc.dbf' SIZE 400G 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 150G AUTOEXTEND ON NEXT 1G
 BLOCKSIZE 16K
 EXTENT MANAGEMENT LOCAL AUTOALLOCATE
 SEGMENT SPACE MANAGEMENT AUTO;

ALTER DATABASE DATAFILE '+DATA/tpcc1/undotbs01.dbf' RESIZE 32760M;
```

Configuring the Oracle pfile

Alter the Oracle pfile on both systems as shown below. We modified the fast_start_mttr_target for the three configurations by changing it to 60, 120, or 180.

```
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='12.1.0.0.0'
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=500g
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=60
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'
```

```
+DATA/orcl/spfileorcl.ora
```

Setting up the HammerDB client

We used a dual-processor server running Red Hat Enterprise Linux 6.5 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 2.16 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 = R920_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
```

Running HammerDB

We ran HammerDB by filling in the appropriate information for the driver options. We tested with a 5-minute ramp up time and 5-minute test duration. We used 101 virtual users with 500-ms user delay and repeat delay. We used rman to back up the database and restore between runs.

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.