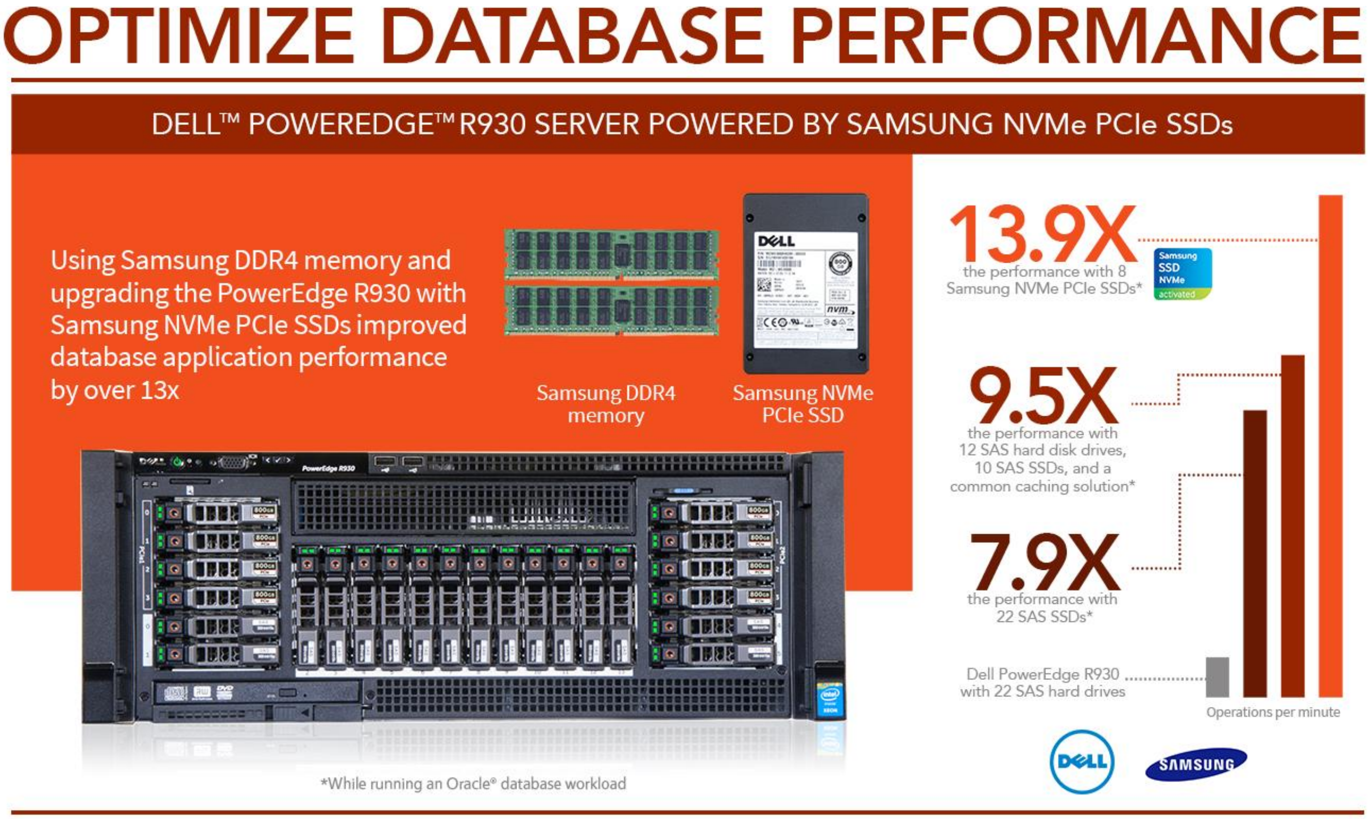
DELL POWEREDGE R930 WITH ORACLE: THE BENEFITS OF UPGRADING TO SAMSUNG NVME PCIE STORAGE



OPTIMIZE DATABASE PERFORMANCE

DELL™ POWEREDGE™ R930 SERVER POWERED BY SAMSUNG NVMe PCIe SSDs

Using Samsung DDR4 memory and upgrading the PowerEdge R930 with Samsung NVMe PCIe SSDs improved database application performance by over 13x

*While running an Oracle® database workload

Responsive and high-performing Oracle database applications, required for smooth business operations, demand high I/O throughput from storage. This means that companies who run these applications need powerful servers with a range of options for upgrading storage. Additionally, the balance of storage and compute performance determines database system performance. Knowing that storage speed and technology is key, Dell designed the new four-socket PowerEdge R930 server with extremely fast storage technology in mind to deliver outstanding performance.

In the Principled Technologies labs, we tested a Dell PowerEdge R930 server running Oracle Database 12*c* with an OLTP workload. We tested the server with four storage configurations: an all-hard disk drive (HDD) configuration, an all-SAS solid-state drive (SSD) configuration, a hybrid configuration using a mix of HDDs and SAS SSDs with a caching solution, and a Samsung-powered NVMe PCIe SSD configuration.

The storage upgrade options of the Dell PowerEdge R930 showed exceptional scalability to meet current and future business demands, delivering up to 13.9 times the already excellent performance of the base configuration. If you are looking for a powerful new server to run your Oracle database infrastructure, consider the new Dell PowerEdge R930 with NVMe PCIe storage for optimal Oracle database performance.

MAY 2015 (Revised)
A PRINCIPLED TECHNOLOGIES TEST REPORT
Commissioned by Dell

UPGRADE STORAGE TO INCREASE PERFORMANCE

The Dell PowerEdge R930 is designed to handle demanding, mission-critical workloads such as enterprise resource planning (ERP), e-commerce, large-scale virtualization, and very large databases. Its standard chassis configuration has twenty-four 2.5-inch drive bays. Additionally, Dell offers a front-loading PCIe SSD chassis with sixteen 2.5-inch drive bays and eight 2.5-inch PCIe SSD drive bays. We tested both server chassis designs using four different storage configurations for the Oracle database files:[1]

- **All-HDD configuration:** 22 SAS HDDs (a standard chassis configuration with twenty-four 2.5-inch drive bays)
- **All-SAS SSD configuration:** 22 SAS SSDs (a standard chassis configuration with twenty-four 2.5-inch drive bays)
- **Hybrid configuration using caching solution:** 10 SAS HDDs and 12 SAS SSDs (a standard chassis configuration with twenty-four 2.5-inch drive bays)
- **Samsung PCIe SSD configuration:** eight Samsung-powered Dell NVMe Express Flash PCIe SSDs (PCIe SSD chassis with sixteen 2.5-inch drive bays and eight 2.5-inch PCIe SSD drive bays)

While hard drives offer the advantages of large capacity and affordability, they may not be able to generate sufficient random I/O throughput and IOPs to keep up with business demands when workloads experience a high level of transactional, random I/O activity. SSDs, whether stand-alone or used as part of a caching solution, can provide that next level of I/O performance for intensive database workloads. When traditional SAS SSDs are not enough, NVMe PCIe SSDs deliver the high I/O performance necessary for highly active workloads.

SAMSUNG TECHNOLOGIES

PCIe NVMe

According to published Samsung specifications, the Samsung NVMe technology, which includes a variety of PCIe-based flash storage, can deliver sequential read speeds of up to 3,000 MB/s and sequential writes up to 2,200 MB/s.[2] Additionally, this technology, present in Dell NVMe Express Flash PCIe SSDs, can support up to 750,000 random read input output operations per second (IOPs) and random writes up to

[1] Note: In all four configurations, we used two hard drives for the operating system.

[2] Throughput based on 3.2 TB NVMe SSD specs (www.samsung.com/global/business/semiconductor/news-events/press-releases/detail?newsId=13701).

130,000 IOPs.[3] More information on Samsung PCIe NVMe technology can be found at www.samsung.com/global/business/semiconductor/product/flash-ssd/nvmessd.

Memory

All configurations in the Dell PowerEdge R930 used Samsung 16GB DDR4 memory. According to Samsung, these modules provide a memory module data transfer rate of up to 2,400 megabits per second per pin, which provides more than 40 percent higher performance, compared to the 1,800Mbps speed of a typical previous-generation DDR3 LRDIMM. Additionally, DDR4 operates at 1.2 volts compared to the typical 1.5 volts for a previous-generation DDR3 LRDIMM, affording up to a 30 percent energy savings. Samsung DDR4 memory is available in 8GB, 16GB, 32GB, 64GB, and 128GB modules. More information on Samsung DDR4 memory can be found at www.samsung.com/global/business/semiconductor/product/computing-dram/overview.

DELL NVME EXPRESS FLASH PCIE SSDS BOOST PERFORMANCE

Dell NVMe Express Flash PCIe solid-state drives, manufactured by Samsung, are designed to deliver lower latency and higher throughput than previous technologies designed for legacy HDDs. PCIe SSDs connect directly to the PCIe bus, which can lower latencies significantly. In addition, the industry standard NVMe protocol is designed to optimize the communications between CPUs and SSDs without legacy HDD protocol overhead.

According to Dell, benefits of these NVMe Express Flash PCIe SSDs include the following:

- Direct PCI bus connection for significantly lower latency and higher I/O
- Industry-standard NVMe queueing to streamline the communication between CPU cores and SSDs (This is a major improvement over legacy queueing designed for HDDs)
- Fast and easy serviceability with hot-swap functions
- High availability for mission-critical workloads
- Enclosure flexibility with a 2.5” form factor
- Excellent power efficiency
- Strong endurance for heavy write workloads

To learn more about Dell NVMe Express Flash PCIe SSDs manufactured by Samsung, visit www.dell.com/learn/us/en/04/campaigns/poweredge-express-flash.

[3] IOPS based on 1.6TB Dell NVMe PCIe SSD specs. See www.samsung.com/global/business/semiconductor/minisite/Greenmemory/news-event/press-release/detail?contentsSeq=13301&contsClassCd=N.

OUR TEST APPROACH AND FINDINGS

While an all-HDD configuration can deliver excellent Oracle database performance, the Dell PowerEdge R930 provides multiple chassis options and storage upgrades that can greatly increase its I/O capabilities. To test this, we configured and ran a heavy OLTP workload on each storage configuration. (For more details on the server configuration, see Appendix A. For the specifics of our testing, see Appendix B.)

Figure 1 shows the relative Oracle database performance of the four configurations we tested. We normalized performance to the all-HDD configuration because Oracle does not permit publishing specific benchmark results.

As Figure 1 shows, replacing all of the HDDs with SAS SSDs delivered 7.9 times the database performance of the all-HDD configuration. The hybrid configuration, using a mix of HDDs and SSDs with a caching solution, delivered 9.5 times the database performance of the base configuration. The final configuration we tested, using eight Samsung NVMe PCIe SSDs, delivered the greatest performance increase—13.9 times the performance of the all-HDD configuration.

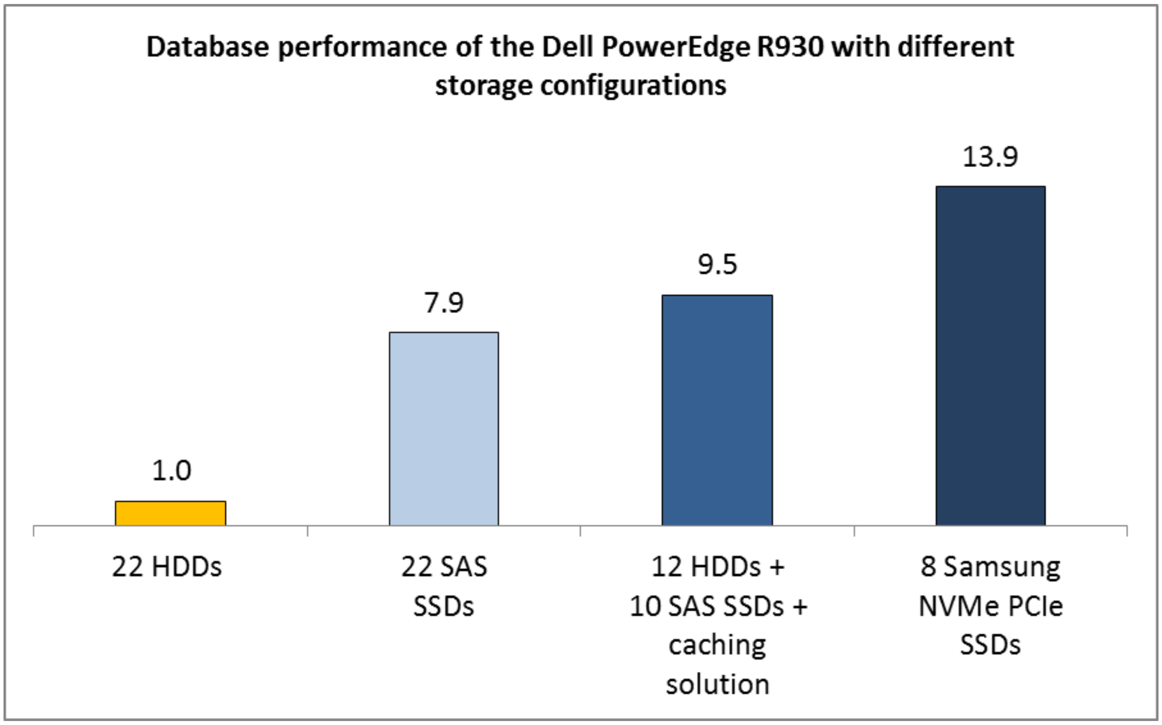


Figure 1: With Samsung NVMe storage configuration upgrades, the Dell PowerEdge R930 supported up to 13.9 times the performance of the all-HDD configuration.

How faster I/O can help your business

The improved storage performance that comes when you upgrade your Oracle database server with faster drives has the potential to benefit your company's return on investment by:

- Enhancing service level agreements
- Lowering database response time and/or supporting more users
- Speeding rebuild times in case of a database crash
- Reducing database maintenance time
- Lowering costs as you eliminate underperforming hardware
- Increasing user satisfaction

CONCLUSION

Strong server performance is essential to companies running Oracle Database. The new Dell PowerEdge R930 provided strong performance with 22 SAS HDDs, but this performance improved when we replaced all of the drives with SAS solid-state drives. It improved further when we used a mix of HDDs and SDDs along with a caching solution. We saw the greatest performance boost when we used eight Samsung-powered NVMe PCIe SSDs. The upgraded configurations of the Dell PowerEdge R930 with Samsung-powered NVMe PCIe SSDs delivered up to 13.9 times the database performance of the all-HDD configuration. This makes the new Dell PowerEdge R930 a powerful platform with scalable storage options that can potentially translate into significant service improvements for your business and your customers, which helps in maximizing ROI.

APPENDIX A–DETAILED SERVER CONFIGURATION

Figure 2 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	0.0.23
BIOS settings	Default
Memory module(s)	
Total RAM in system (GB)	256
Vendor and model number	Samsung® M393A2G40DB0-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)	16
Number of RAM module(s)	16
Chip organization	Double-sided
Rank	Dual

System	**Dell PowerEdge R930**
Operating system	
Name	Red Hat® Enterprise Linux®
Kernel	2.6.32-431.el6.x86_64
Language	English
RAID controller	
Vendor and model number	PERC H730P
Firmware version	25.2.1.0037
Cache size (MB)	2,048
Local storage	
Hard drive #1	
Vendor and model number	Seagate® ST9300653SS
Number of drives	22 (database and logs in HDD testing)
Size (GB)	300
RPM	15,000
Type	SAS
Hard drive #2	
Vendor and model number	Seagate ST9900805SS
Number of drives	14 (2 for OS, 12 used for database and logs in DAS cache testing)
Size (GB)	900
RPM	10,000
Type	SAS
SAS solid-state drive	
Vendor and model number	Dell 2XR0K
Number of drives	22 (database and logs in all SSD testing)
Size (GB)	200
RPM	N/A
Type	SAS SSD
PCIe solid-state drive	
Vendor and model number	Dell MZ-WEI8000 (Samsung NVMe PCIe)
Number of drives	8 (PCIe SSD testing)
Size (GB)	800
RPM	N/A
Type	NVMe PCIe SSD
Ethernet adapters	
Vendor and model number	Intel Gigabit 4P i350-t
Type	Gigabit Ethernet

Figure 2: Detailed configuration information for the test configurations.

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 (OLTP workload derived from TPC-C) workload and a data warehouse (workload derived from TPC-H) workload. For this study, we used the transactional workload. This transactional workload is derived from TPC-C, and as such is not comparable to published TPC-C results. For more information about HammerDB, visit hammerora.sourceforge.net.

Configuring disk drives

We designed each configuration to provide the best possible performance while staying as close as possible to the Oracle best practices guide. We used two different servers for the test. For the all-HDD, all-SAS SSD, and hybrid drive with DAS Cache configurations, we used a Dell PowerEdge R930 with 24 SAS hard drive bays. For the PCIe SSD configuration, we used a Dell PowerEdge R930 with 16 SAS hard drive bays and eight PCIe SSD bays. We configured both servers identically except for the hard drive bays. We provide the details of each configuration below.

All-HDD configuration

We used two 900 GB 10,000 RPM SAS hard drives configured in a RAID 1 to hold the OS. We used 22 300GB 15,000 RPM SAS for database and logs. We configured the 22 hard drives in 11 RAID 1 with write back policy. We used 10 of the RAID 1 for the database and one RAID 1 for the logs.

All-SAS SSD configuration

We used two 900 GB 10,000 RPM SAS hard drives configured in a RAID 1 to hold the OS. We used 22 200GB SAS SSDs for database and logs. We configured the 22 SAS SSDs in 11 RAID 1 with write through policy. We used 10 of the RAID 1 for the database and one RAID 1 for the logs.

Hybrid configuration with caching solution

We used two 900 GB 10,000 RPM SAS hard drives configured in a RAID 1 to hold the OS. We used 12 900GB 10,000 RPM SAS hard drives for the database. We configured the 12 drives into four three-disk RAID 5 configurations with write back policy. We used two 200GB SAS SSD for logs configured in a RAID 1 with write through policy. We used eight SAS SSDs for our caching solution, SanDisk DAS Cache. We configured the eight drives into four RAID 1 volumes with write through policy. We configured the SanDisk DAS Cache so each of the four RAID 5 volumes used one SSD RAID 1 for the acceleration.

Samsung PCIe SSD configuration

We used two 900 GB 10,000RPM SAS hard drives configured in a RAID 1 to hold the OS. We used eight 800 GB NVMe PCIe SSDs for the database and logs. We presented all eight drives to Oracle ASM and configured them as one volume for testing. We used ASM redundancy because the PCIe drives are not on a RAID controller.

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 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 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 = 135382798336
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 102400
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
```

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       R930 R930.localhost.localdomain localhost
localhost.localdomain localhost4 localhost4.localdomain4
::1             R930 R930.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=$TMP
export ORACLE_HOSTNAME=R930.localhost.localdomain
export ORACLE_BASE=/home/oracle/app/oracle
export GRID_HOME=$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'
```

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

   `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}=="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"
```

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 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
```

Setting up the DAS Cache storage

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

1. Add the cache device in write-back caching mode:

```
fscli --add-cache /dev/sdb1
```

2. List the cache device to get cache tag to be used later:

```
fscli --list-cache
```

3. Determine the name of the block device to be accelerated:

```
ls /dev/fio
```

4. Make sure that the data volume to be accelerated is not currently mounted:

```
ls -al /dev/disk/by-id/scsi*
```

5. Add the data volume that is going to be accelerated:

```
fscli --add-vol /dev/disk/by-id/scsi-36d4ae52000948b63000035ea534f6e5e
```

6. Enable caching:

```
fscli --enable-accel dev cache-tag=tag
```

7. Confirm volume is accelerated:

```
ls -l /dev/fio
```

8. 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}=="36c81f660d90093001cc
968be0ae9269a", SYMLINK+="oracleasm/logs", OWNER="oracle", GROUP="dba",
MODE="06
60"
   SYMLINK=="fio/disk-by-id-scsi-36c81f660d90093001cc9686c05fe5bdd",
   SYMLINK+="orac
```

```
leasm/data1", OWNER="oracle", GROUP="dba", MODE="0660"
SYMLINK=="fio/disk-by-id-scsi-36c81f660d90093001cc968820758ed5c",
SYMLINK+="orac
leasm/data2", OWNER="oracle", GROUP="dba", MODE="0660"
SYMLINK=="fio/disk-by-id-scsi-36c81f660d90093001cc9689908b41ba5",
SYMLINK+="orac
leasm/data3", OWNER="oracle", GROUP="dba", MODE="0660"
SYMLINK=="fio/disk-by-id-scsi-36c81f660d90093001cc968aa09ba5325",
SYMLINK+="orac
leasm/data4", OWNER="oracle", GROUP="dba", MODE="0660"
```

9. Execute udevadm and start udev:

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

10. List the ASM devices:

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

```
lrwxrwxrwx 1 root root 8 Apr 23 15:33 data1 -> ../fio-3
lrwxrwxrwx 1 root root 8 Apr 23 15:33 data2 -> ../fio-4
lrwxrwxrwx 1 root root 8 Apr 23 15:33 data3 -> ../fio-5
lrwxrwxrwx 1 root root 8 Apr 23 15:33 data4 -> ../fio-6
lrwxrwxrwx 1 root root 7 Apr 23 15:33 logs -> ../sdf1
```

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 HDD, all SAS SSD and DAS Cache configurations, 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.

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
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 all drives, and Click Next. (We did not select the last disk on configuration. We configured the last drive for logs.) Select External 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. 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.

Configure disks for log files

We used the steps below 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*

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 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 --

-- BEGIN: SSD REDO LOGS --

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;
-- END: SSD REDO LOGS --

-- BEGIN: SAS REDO LOGS --
ALTER DATABASE ADD LOGFILE GROUP 1 ( '+LOGS/orcl/redo01.log' ) SIZE 50G;
ALTER DATABASE ADD LOGFILE GROUP 2 ( '+LOGS/orcl/redo02.log' ) SIZE 50G;
ALTER DATABASE ADD LOGFILE GROUP 3 ( '+LOGS/orcl/redo03.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

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/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='12.1.0.0.0'
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=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 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 = 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
```

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.

APPENDIX C – WHAT WE TESTED

About the Dell PowerEdge R930

The Dell PowerEdge R930 is part of Dell's 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 eight 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 Samsung NVMe PCIe SSDs

The newly developed high-speed NVMe SSD provides a sequential read speed up to 3,000MB/s, which allows it to process 500GB of data (equivalent to 100 Full HD movies 5GB in length) in less than 3 minutes. This new SSD is 14 times faster than a high-end enterprise HDD, and six times faster than Samsung's former high-end enterprise SSD storage. The random read performance reaches up to 750,000 IOPS (Input Output Operations Per Second), which is more than 10 times as fast as most other high-end SSD storage options.

To learn more about Dell NVMe Express Flash PCIe SSDs manufactured by Samsung, visit www.dell.com/learn/us/en/04/campaigns/poweredge-express-flash.

About the caching software

The caching solution we used in the hybrid configuration of the Dell PowerEdge R930, SanDisk DAS Cache, helps accelerate the speed of storage input-output (I/O) operations, and therefore improves the performance of I/O-intensive applications such as database applications.

The caching solution leverages SSDs on the host server to create a read-write cache for the most frequently accessed or "hot" data. Typically, a minority of the active dataset accounts for 70 percent to 80 percent of server I/O. Accelerating this hot data on a SSD-based cache tier can deliver much greater performance than using hard disk drives.

The caching solution is transparent to existing software layers and requires no changes to running applications or the underlying storage configuration.

Learn more at www.dell.com/learn/us/en/04/server-technology-components-caching

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.