



The science behind the report:

Get stronger Oracle Database OLTP performance with the Dell EMC PowerEdge R6525

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report [Get stronger Oracle Database OLTP performance with the Dell EMC PowerEdge R6525](#).

We concluded our hands-on testing on June 4, 2020. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on May 27, 2020 or earlier. Unavoidably, these configurations may not represent the latest versions available when we publish this report.

Our results

We ran the workload three times on each server and selected the median result of the three runs. Note that because the Oracle Database EULA does not permit us to publish exact results, we normalized performance numbers between the two platforms to make our comparison.

Table 1: Results of our testing on a Dell EMC™ PowerEdge™ R6525 server with AMD EPYC™ 7F72 processors vs. an HPE ProLiant DL360 Gen10 server with Intel® Xeon® Gold 6248R processors.

	Dell EMC PowerEdge R6525	HPE ProLiant DL360 Gen10
Transactions per minute (median result, normalized)	1.51	1.00
Hardware cost (USD)	\$49,417.01*	\$52,701.00**

*We obtained this information from the Dell EMC website on June 8, 2020.

**We obtained this information from a certified HPE reseller on June 8, 2020.

Bill of materials

Table 2: Bill of materials for the Dell EMC PowerEdge R6525 we tested.

Product description	Product number	Quantity
PowerEdge R6525 server	G0EY42	1
AMD EPYC 7F72 3.30GHz, 24C/48T, 192M cache (240W) DDR4-3200	GI6GQ2L	1
Chassis with up to 10x2.5" drives	GC6DRSG	1
No Trusted Platform Module	GMHJL5Y	1
Chassis with up to 10 x 2.5" hot plug NVMe drives (2CPU, No XGMI & CPU, 180W or More) V2	G65QIZX	1
PowerEdge R6525 shipping	G1D0638	1
PowerEdge R6525 shipping material for x4/10 chassis	GMTX7Q9	1
AMD EPYC 7F72 3.30GHz, 24C/48T, 192M cache (240W) DDR4-3200	GEDPTA4	1
Heatsink for 2 CPU configuration (CPU greater than or equal to 180W)	GCWJ8LK	1
NVMe backplane	GL6C9WE	1
3200MT/s RDIMMs	GR3CFNV	1
Performance Optimized	GH9QBEI	1
32GB RDIMM, 3200MT/s, Dual Rank	GQJ1WLK	16
C30, No RAID for NVME chassis	GF5RC29	1
No controller	GLTBAZ3	1
No hard drive	GZ4G6T3	1
BOSS controller card + with 1 M.2 sticks 240G (No RAID), LP	G052ACP	1
Dell 1.6TB, NVMe, Mixed Use Express Flash, 2.5 SFF drive, U.2, PM1725b with carrier	G402CN1	6
No operating system	G78MU35	1
iDRAC9, Express X5	GXWHJR2	1
iDRAC, Factory Generated Password	G2T768J	1
Riser Config 2, 1 x 16 Gen4 LP PCIe slot (CPU1), 2 x 16 LP PCIe slot (CPU2)	GRVK2HX	1
Broadcom 57414 Dual Port 10/25GbE SFP28, OCP NIC 3.0	G63VWGR	1
Very high performance fan for 10 x 2.5" or 10 x 2.5 + 2 chassis	GR2SJ86	1
Dual, hot-plug, redundant power supply (1+1), 1400W	GIW6SNG	1
NEMA 5-15P to C13 wall plug, 125 Volt, 15 AMP, 10 feet (3m), power cord, North America	G749Q3L	2
No bezel for x8 and x10 chassis	GZ3N9YG	1
No Quick Sync	GI1FEZA	1
Performance BIOS setting	GJO594B	1
UEFI BIOS boot mode with GPT partition	GSFTG4Y	1
ReadyRails sliding rails with cable management arm	GQLCAO2	1
No systems documentation, No OpenManage DVD kit	GVRYSM7	1

Product description	Product number	Quantity
PowerEdge R6525 Motherboard, with 2 x 1Gb onboard LOM (BCM5720)	G1UPED3	1
Basic next business day 36 Months, 36 Month(s)	G5ZBAK0	1
No installation	NOINSTL	1

Table 3: Bill of materials for the HPE ProLiant DL360 Gen10 we tested.

Product description	Product number	Quantity
HPE ProLiant DL360 Gen10 Premium 10NVMe NC configure-to-order server	P19767-B21	1
HPE DL360 Gen10 10SFF ModX CTO	P19767-B21 ABA	1
Intel Xeon-Gold 6248R (3.0GHz/24-core/205W) FIO Processor Kit for HPE ProLiant DL360 Gen10	P24487-L21	1
Intel Xeon-Gold 6248R (3.0GHz/24-core/205W) Processor Kit for HPE ProLiant DL360 Gen10	P24487-B21	1
Factory integrated	P24487-B21 OD1	1
HPE 32GB (1x32GB) Dual Rank x4 DDR4-2933 CAS-21-21-21 Registered Smart Memory Kit	P00924-B21	16
Factory integrated	P00924-B21 OD1	16
HPE 1.6TB NVMe x4 lanes mixed use SFF (2.5in) SCN 3yr Wty Digitally Signed Firmware SSD	P10222-B21	6
Factory integrated	P10222-B21 OD1	6
HPE 960GB SATA 6G Mixed Use SFF (2.5in) SC 3yr Wty SSD	P19949-B21	1
Factory integrated	P19949-B21 OD1	1
HPE Ethernet 1Gb 4-port 331T adapter	647594-B21	1
Factory integrated	647594-B21 OD1	1
HPE 96W Smart Storage lithium-ion battery with 145mm cable kit	P01366-B21	1
Factory integrated	P01366-B21 OD1	1
HPE Smart Array P408i-a SR Gen10 (8 internal lanes/2GB cache) 12G SAS modular LH controller	869081-B21	1
Factory integrated	869081-B21 OD1	1
HPE Ethernet 10/25Gb 2-port 640FLR-SFP28 adapter	817749-B21	1
Factory integrated	817749-B21 OD1	1
HPE 800W Flex Slot Platinum hot plug low halogen power supply kit	865414-B21	2
Factory integrated	865414-B21 OD1	2
HPE iLO Advanced 1-server License with 3yr Support on iLO Licensed Features	BD505A	1
Factory integrated	BD505A OD1	1
HPE 1U Gen10 SFF Easy Install Rail Kit	874543-B21	1
Factory integrated	874543-B21 OD1	1
HPE 3Y Foundation Care 24x7 SVC	H7J34A3	1
HPE DL360 Gen10 Support	H7J34A3 WAG	1

System configuration information

Table 4: Detailed information on the systems we tested.

Server configuration information	Dell EMC PowerEdge R6525	HPE ProLiant DL360 Gen10
BIOS name and version	Dell 1.2.11	U32 v2.34
Non-default BIOS settings	N/A	N/A
Hyperthreading	Enabled	Enabled
Operating system name and version/ build number	Red Hat® Enterprise Linux® 7.8 Kernel: 3.10.0-1127.8.2.el7.x86_64	Red Hat Enterprise Linux 7.8 Kernel: 3.10.0-1127.8.2.el7.x86_64
Date we applied last OS and firmware updates	5/14/2020	5/14/2020
Power management policy	System Profile = Performance	Workload Profile = Virtualization - Max Performance
Processor		
Number of processors	2	2
Vendor and model	AMD EPYC 7F72	Intel Xeon Scalable Gold 6248R
Core count (per processor)	24	24
Core frequency (GHz)	3.20	3.00
Stepping	0	B1
Memory module(s)		
Total memory in system (GB)	512	512
Number of memory modules	16	16
Vendor and model	Hynix HMA84GR7CJR4N-XN	Hynix HMA84GR7CJR4N-WM
Size (GB)	32	32
Type	PC4-3200	PC4-2933Y
Speed (MHz)	3,200	2,933
Speed running in the server (MHz)	3,200	2,933
Storage controller		
Vendor and model	Dell HBA345	HPE Smart Array P408i-a SR Gen10
Cache size (GB)	N/A	2
Firmware version	11.08.06.00	1.99
Local storage (type A)		
Number of drives	1	1
Drive vendor and model	Intel SSDSCKKB240G8R	Samsung-MZ-7LM960B
Drive size (GB)	240	960
Drive information (speed, interface, type)	6 Gb/s, M.2 SATA, SSD	6 Gb/s, SATA, SSD

Server configuration information	Dell EMC PowerEdge R6525	HPE ProLiant DL360 Gen10
Local storage (type B)		
Number of drives	6	6
Drive vendor and model	Samsung® PM1725b	Samsung PM1725b
Drive size (GB)	1,600	1,600
Drive information (speed, interface, type)	NVMe, PCIe SSD	NVMe, PCIe SSD
Network adapter		
Vendor and model	Broadcom® BCM57414 NetXtreme®-E 10Gb/25Gb RDMA Ethernet Controller	HPE 640FLR-SFP28 Adapter
Number and type of ports	2 x 10Gb/25Gb	2 x 10Gb/25Gb
Driver version	1.0.1	14.24.9006
Cooling fans		
Vendor and model	Dell PIH040M12P	HPE 873799-001
Number of cooling fans	4	7
Power supplies		
Vendor and model	Dell A1400E-S0	HPE HSTNS-PL41-1
Number of power supplies	2	2
Wattage of each (W)	1,400	800

About our testing

Our hands-on testing compared the following solutions:

- A dual-socket Dell EMC PowerEdge R6525 powered by AMD EPYC 7F72 processors. The list price of this configuration as of June 8, 2020 was \$49,417.01. We obtained this information from the Dell EMC website.
- A dual-socket HPE ProLiant DL360 Gen10 powered by Intel Xeon Gold 6248R processors. The list price of this configuration as of June 8, 2020 was \$52,701.00. We obtained this information from a certified HPE reseller.

We set out to measure the performance capabilities of the two servers running transactional database workloads. We selected the AMD EPYC 7F72 processors from the newly released 2nd Gen EPYC line. Similarly, we selected the Intel Xeon Gold 6248R from the newly released Cascade Lake refresh and matched the number of cores with the AMD EPYC 7F72.

We configured each system with 512 GB of RAM at the maximum supported speed for each configuration (3200 MHz for AMD and 2933 MHz for Intel), and six NVMe PCIe SSDs for the database data and log files. We installed Red Hat Enterprise Linux 7.8 and the Oracle 19c grid infrastructure on both systems. We used Oracle ASM to create two-way mirrored diskgroups with the NVMe PCIe SSDs, with four drives in a disk group for the data, and the remaining two drives in a second diskgroup for the logs. After configuring the ASM disk groups, we installed the Oracle Database 19c software and used the database configuration assistant to create our database. After we had deployed our database, we recreated the redo logs to make them larger and then created the tablespaces necessary for HammerDB to load the data into. We then restarted the database with the spfile on [page 13](#). We set up a HammerDB client on a separate server running CentOS 7 and measured the transactional database performance of each system running a TPC-C-like workload, which targets a transactional database and produces a metric of transactions per minute. We used HammerDB to generate a 3,750-warehouse TPC-C-like database schema that roughly amounts to 300 GB in size. After creating the schema, in HammerDB, we generated 384 virtual users to target the two databases and run the TPC-C-like workload. With this configuration, we achieved an average CPU utilization of above 80 percent on each system, with both reaching a peak of 97 percent during the run. We ran our tests with a ramp-up time of 15 minutes and a runtime of 30 minutes. We note the test parameter changes we made in the Running the TPC-C-like test section of this . For all other parameters, we used the default.

We used list pricing to calculate a price-to-performance ratio for each solution.

Installing Red Hat Enterprise Linux 7.8

1. Boot to the Red Hat Enterprise Linux 7.8 installation media.
2. Select Install or upgrade an existing system.
3. Choose English, and click Continue.
4. Under Installation Destination, select the desired disk to install the OS.
5. Under Other Storage Options, select I will configure partitioning, and click Done.
6. Select Click here to create them automatically.
7. Remove the /home partition if it exists.
8. Expand the swap partition to 32GB.
9. Assign all remaining free space to the / partition.
10. Click Done.
11. Click Accept Changes.
12. Select Kdump.
13. Uncheck Enable kdump, and click Done.
14. Select Network & Hostname.
15. Enter the desired hostname for the system.
16. Turn on the desired network ports, and click Configure.
17. On the General tab, select Automatically connect to this network when it is available.
18. On the IPv4 Settings tab, choose the drop-down named Method, and select Manual.
19. Under Addresses, click Add, and enter the desired static IP information for the server.
20. Enter the desired DNS information.
21. Click Save, and click Done.
22. Select Date & Time, and ensure you have set the correct date, time, and time zone.
23. To add your NTP server, click the cog next to the Network Time On/Off switch.
24. Add the IP address of your NTP server, and click +.
25. Uncheck all other NTP servers.
26. Click OK.
27. Click Done.
28. Click Software Selection.
29. Choose the Base Environment of Minimal Install.

30. Click Done.
31. Click Begin Installation.
32. Select Root Password.
33. Enter the desired root password, and click Done.
34. When the installation completes, to restart the server, select Reboot.

Configuring Red Hat Enterprise Linux 7.8 for Oracle

1. Log onto the server as root.
2. Disable the firewall:


```
systemctl stop firewalld
systemctl disable firewalld
```
3. Disable SELinux:


```
vi /etc/selinux/config
SELINUX=disabled
```
4. Register your RHEL subscription:


```
subscription-manager register --username <RHEL_USERNAME> --password <RHEL_PASSWORD> --auto-attach
```
5. Install wget, sysstat, and gdisk:


```
yum install -y sysstat wget gdisk
```
6. Update RHEL 7.8:


```
yum update -y
```
7. Disable auditd:


```
systemctl disable auditd
```
8. Download the Oracle Database 19c preinstall RPM.


```
curl -o oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm https://yum.oracle.com/repo/OracleLinux/OL7/latest/x86_64/getPackage/oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm
```
9. Download the compat-libstdc++-33-3.2.3-72.el7.x86_64.rpm package:


```
wget http://mirror.centos.org/centos/7/os/x86_64/Packages/compat-libstdc++-33-3.2.3-72.el7.x86_64.rpm
```
10. Install compat-libstdc++-33:


```
yum localinstall -y compat-libstdc++-33-3.2.3-72.el7.x86_64.rpm
```
11. Install the Oracle preinstall:


```
yum localinstall -y oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm
```
12. Disable transparent hugepages and re-enable numa persistently by adding "transparent_hugepage=never" and removing "numa=off" from the end of the GRUB_CMDLINE_LINUX option in /etc/default/grub and running the following command to rebuild the grub.cfg file:


```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```
13. Reboot.
14. Create groups for ASM by running these shell commands:


```
groupadd -g 54327 asmdba
groupadd -g 54328 asmoper
groupadd -g 54329 asmadmin
```
15. Modify the oracle user to add the ASM groups:


```
usermod -G asmdba,asmoper,asmadmin -a oracle
```
16. Create passwords for the Oracle account with passwd:


```
passwd oracle
```

17. On a separate computer, navigate to the RHEL Labs page for the RHEL Tuner script for Oracle (<https://access.redhat.com/labs/rhel-tfo/>), and fill out the proper details for your system (we put the SGA as 384GB). Click the button to generate the script for your server.

- Red Hat Enterprise Linux Version: Red Hat Enterprise Linux 7
- Oracle Group Name/ID: 54321
- System Global Area (SGA): 384 GB
- Maximum number of open file descriptors: 65536 (Default)

18. Copy the RHEL Tuner script to your server.

19. Change the permissions of the RHEL Tuner script to make it executable:

```
chmod +x rhel-config-for-oracle.sh
```

20. On your server, accept all changes, and execute the RHEL Tuner script:

```
./rhel-config-for-oracle.sh
```

21. Create the following directories, and assign the following permissions:

```
mkdir -p /u01/app/19.0.0.0/grid
mkdir -p /u01/app/oracle/product/19.0.0.0/dbhome_1
chown -R oracle:oinstall /u01
chmod -R 775 /u01/
```

22. Append the following to the `/etc/security/limits.conf`:

```
oracle - nofile 65536
oracle - nproc 16384
oracle - stack 32768
oracle - memlock 134217728
soft memlock unlimited
hard memlock unlimited
```

23. Log into the Oracle user:

```
su - oracle
```

24. Add the following lines to the `.bash_profile`:

```
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=<HOSTNAME>
export ORACLE_UNQNAME=orcl
export ORACLE_BASE=/u01/app/oracle
export GRID_HOME=/u01/app/19.0.0.0/grid
export DB_HOME=$ORACLE_BASE/product/19.0.0.0/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'
```

25. Create the following files in the Oracle user's home folder:

```
>>>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:$ORACLE_HOME/rdbms/jlib
>>>db_env<<<
export ORACLE_SID=orcl
export ORACLE_HOME=$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
```

26. As the root user, install the ASMLib kernel module package:

```
yum install -y kmod-oracleasm
```


27. Download the OracleASM support (oracleasm-support-2.1.11-2.el7.x86_64.rpm) and lib (oracleasm-lib-2.0.12-1.el7.x86_64.rpm) packages from here: <https://www.oracle.com/linux/downloads/linux-asmlib-rhel7-downloads.html>.
28. Copy the packages to your RHEL server.
29. From the directory you copied the packages to, install the Oracle ASM lib and support packages:

```
yum localinstall -y oracleasm*
```

30. Edit /etc/sysconfig/oracleasm to contain the following:

```
# ORACLEASM_ENABLED: 'true' means to load the driver on boot.
ORACLEASM_ENABLED=true
# ORACLEASM_UID: Default UID owning the /dev/oracleasm mount point.
ORACLEASM_UID=oracle
# ORACLEASM_GID: Default GID owning the /dev/oracleasm mount point.
ORACLEASM_GID=oinstall
# ORACLEASM_SCANBOOT: 'true' means fix disk perms on boot
ORACLEASM_SCANBOOT=true
# ORACLEASM_USE_LOGICAL_BLOCK_SIZE: 'true' means use the logical block
# size reported by the underlying disk instead of the physical. The
# default is 'false' ORACLEASM_USE_LOGICAL_BLOCK_SIZE=false
```

31. Run the following command to initialize oracleasm:

```
oracleasm init
```

32. Create a partition on all disks using gdisk.
33. Run the following commands to configure all of the disks for Oracle ASM:

```
oracleasm createdisk DATA1 /dev/<DISK1>
oracleasm createdisk DATA2 /dev/<DISK2>
oracleasm createdisk DATA3 /dev/<DISK3>
oracleasm createdisk DATA4 /dev/<DISK4>
oracleasm createdisk LOG1 /dev/<DISK5>
oracleasm createdisk LOG2 /dev/<DISK6>
```

Installing Oracle Grid Infrastructure 19c

1. Log in as the Oracle user.
2. To set the Oracle grid environment, type `grid_env`
3. Change directories to the grid home directory:


```
cd $ORACLE_HOME
```
4. Unzip `linuxx64_193000_grid_home.zip`:


```
unzip -q <grid_zip_location>/LINUX.X64_193000_grid_home.zip
```
5. To start the installer, type `./gridSetup.sh`
6. In the Select Installation Option screen, select Install and Configure Grid Infrastructure for a Standalone Server, and click Next.
7. Choose the language, and click Next.
8. In the Create ASM Disk Group screen, choose the Disk Group Name (DATA), and change redundancy to Normal.
9. Change the path to `/dev/oracleasm/disks`, and select the DATA disks.
10. In the Specify ASM Password screen, choose Use same password for these accounts, write the passwords for the ASM users, and click Next.
11. At the Management Options screen, click Next.
12. Leave the default Operating System Groups, and click Next.
13. Leave the default installation, and click Next.
14. Leave the default inventory location, and click Next.
15. Under Root script execution, leave the defaults, and click Next.
16. On the Prerequisite Checks screen, make sure that there are no errors.
17. On the Summary screen, verify that everything is correct, and click Finish to install Oracle Grid Infrastructure.
18. At one point during the installation, the installation prompts you to execute a configuration script as root. Follow the instructions to run the scripts.
19. At the Finish screen, click Close.
20. To run the ASM Configuration Assistant, type `asmca`
21. In the ASM Configuration Assistant, click Create.

22. In the Create Disk Group window, name the new disk group LOG, choose redundancy Normal Redundancy, and select the two LOG disks.
23. Click Advanced Options.
24. Set the database compatibility level to 19.0.0.0.0, and click OK.
25. Exit the ASM Configuration Assistant.

Installing Oracle Database 19c

1. To set the Oracle database environment, type `db_env`
2. Change directories to the database home directory:

```
cd $ORACLE_HOME
```
3. Unzip LINUX.X64_193000_db_home.zip:

```
unzip -q <database_zip_location>/LINUX.X64_193000_db_home.zip
```
4. Run `./runInstaller`.
5. Wait for the GUI installer to load.
6. At Select Configuration Option, select Set Up Software Only, and click Next.
7. At Select Database Installation Option, select Single instance database installation, and click Next.
8. At Select Database Edition, select Enterprise Edition, and click Next.
9. At Specify Installation Location, leave the default Oracle base, and click Next.
10. At Privileged Operating System groups, keep default, and click Next.
11. At Perform Prerequisite Checks, wait for the verification results, fix all errors, and click Next.
12. Review Summary page, and click Install.
13. Once the Execute Configuration scripts prompt appears, ssh into the server as root, and run the following command:

```
/u01/app/oracle/product/19.0.0.0/dbhome_1/root.sh
```
14. Return to the prompt, and click OK.
15. Once the installer completes, click Close.

Creating and configuring the database

1. Using SSH -Y, log into the host.
2. To switch to the first database environment, type `db_env`
3. Type `dbca`. To open the Database configuration assistant, press Enter.
4. At the Database Operation screen, select Create Database, and click Next.
5. Under Creation Mode, select Advanced configuration, and click Next.
6. At the Database Deployment Type screen, select General Purpose. Click Next.
7. Enter a Global database name and the appropriate SID, and uncheck Create as Container database. Click Next.
8. At the storage option screen, select Use following for the database storage attributes.
9. In the drop-down menu, select Automatic Storage Management (ASM), select +DATA for the file location, and click Next.
10. At the Fast Recovery Option screen, check the box for Specify Fast Recovery Area.
11. In the drop-down menu, select ASM, select +DATA for the Fast Recovery Area, and enter 500GB for the size.
12. At the Specify Network Configuration Details screen, select the listener, and click Next.
13. At the Specify Oracle Data Vault Config Option screen, leave the default options, and click Next.
14. At the Specify Configuration Options screen, leave the memory selections to defaults, and click Next.
15. At the Specify Management Options screen, select Configure Enterprise Manager (EM) Database Express, and click Next.
16. At the User Credentials screen, select Use the same administrative password for all accounts, enter and confirm the desired password, and click Next.
17. At the Creation Options screen, select Create Database, and click Next.
18. At the summary screen, click Finish.
19. Close the Database Configuration Assistant.

Configuring Oracle Tablespaces and redo log

In this section, we configured the database so that we could load the data with HammerDB and run the tests. We started by creating temporary log groups so we could drop the default log groups and files. After creating temporary log groups and log files, we dropped the defaults and recreated two new redo log groups and files with the size set to 400G. We then altered the system settings to allow us to create files with block sizes other than the default 8K. We created a TPCC tablespace and a TPCC_OL tablespace for the order lines table. Finally, we resized the existing undo tablespace file and added three more files to the undo tablespace.

Alter the tablespaces on both systems as shown below. To enter SQL prompt, type 'sqlplus / as sysdba':

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

ALTER DATABASE ADD LOGFILE GROUP 1 ( '+LOG/redo01.log' ) SIZE 400G
BLOCKSIZE 4K;
ALTER DATABASE ADD LOGFILE GROUP 2 ( '+LOG/redo02.log' ) SIZE 400G
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;

SHUTDOWN

STARTUP

HOST rm -f /tmp/temp*.log

CREATE BIGFILE TABLESPACE "TPCC"
DATAFILE '+DATA/ORCL/DATAFILE/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/DATAFILE/tpcc_ol.dbf' SIZE 200G AUTOEXTEND ON NEXT 1G
BLOCKSIZE 16K
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;

ALTER DATABASE DATAFILE '+DATA/ORCL/DATAFILE/<UNDO FILE>' RESIZE 32760M;
ALTER TABLESPACE UNDOTBS1 ADD DATAFILE '+DATA/ORCL/DATAFILE/undo2.dbf' SIZE 32760M MAXSIZE UNLIMITED;
ALTER TABLESPACE UNDOTBS1 ADD DATAFILE '+DATA/ORCL/DATAFILE/undo3.dbf' SIZE 32760M MAXSIZE UNLIMITED;
ALTER TABLESPACE UNDOTBS1 ADD DATAFILE '+DATA/ORCL/DATAFILE/undo4.dbf' SIZE 32760M MAXSIZE UNLIMITED;
```

Configuring CentOS 7 for HammerDB client

On our client system, we installed CentOS 7 and configured it for the Oracle Database 19c Client and HammerDB. Note: To perform the 'yum ...' commands in this section, the server must have external connectivity to <http://mirror.centos.org>.

1. Log onto the server as root.
2. Disable the firewall:

```
systemctl stop firewalld
systemctl disable firewalld
```
3. Disable SELinux:

```
vi /etc/selinux/config
SELINUX=disabled
```
4. Install sysstat, gcc, gcc-c++:

```
yum install -y sysstat gcc gcc-c++
```
5. Update CentOS 7:

```
yum update -y
```
6. Disable auditd:

```
systemctl disable auditd
```
7. Reboot.
8. Download the Oracle Database preinstall RPM:

```
curl -o oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm https://yum.oracle.com/repo/OracleLinux/OL7/latest/x86_64/getPackage/oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm
```
9. Install compat-libstdc++-33:

```
yum install -y compat-libstdc++-33
```
10. Install the Oracle Database preinstall RPM:

```
yum localinstall -y oracle-database-preinstall-19c-1.0-1.el7.x86_64.rpm
```
11. Create a password for the Oracle user with passwd:

```
passwd oracle
```
12. Log into the Oracle user with X11 forwarding:

```
ssh -Y oracle@<IP_ADDRESS>
```
13. Unzip LINUX.X64_193000_client.zip:

```
unzip -q LINUX.X64_193000_client.zip
```
14. Change directories into the unzipped folder:

```
cd client/
```
15. Run the client installer:

```
./runInstaller
```
16. At the Select Installation Type screen, select Administrator, and click Next.
17. Leave the defaults for the install location, and click Next.
18. Leave the defaults for the inventory directory and group name, and click Next.
19. At the prerequisite check screen, fix any prerequisite errors, and continue.
20. Click Install.
21. When prompted, run the root script as the root user:

```
/home/oracle/app/oraInventory/orainstRoot.sh
```
22. Click OK.
23. Click Close.
24. Download the tarball of the latest version of HammerDB.
25. Install HammerDB:

```
tar xfv HammerDB-3.3-Linux.tar.gz
```

26. To point HammerDB to the Oracle libraries, add the following lines to the `.bash_profile`:

```
export ORACLE_BASE=/home/oracle/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/19.0.0/client_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export ORACLE_LIBRARY=$LD_LIBRARY_PATH/libclntsh.so
export PATH=$ORACLE_HOME/bin:$PATH
export ORACLE_SID=orcl
```

27. Change directories to the tns admin directory:

```
cd /home/oracle/app/oracle/product/19.0.0/client_1/network/admin/
```

28. Copy the `tnsnames.ora` file from the sample directory below to the current directory:

```
cp samples/tnsnames.ora ./
```

29. Edit the `tnsnames.ora` file, and add the following lines to configure the instance connection:

```
ORCL=(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=orcl)) (ADDRESS=(PROTOCOL=tcp) (HOST=<IP_ADDRESS>) (PORT=1521)))
```

30. Test the connection to make sure you can ping the Oracle server with the ORCL service name:

```
tnsping ORCL
```

Installing Docker CE on HammerDB client

1. Log into the HammerDB CentOS 7 client as the root user.

2. Uninstall any previous versions of Docker-Engine:

```
yum remove docker docker-common docker-selinux docker-engine
```

3. Install prerequisites:

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

4. Setup stable repo:

```
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

5. Install Docker CE:

```
yum -y install docker-ce docker-ce-cli containerd.io
```

6. Start and enable docker service:

```
systemctl start docker && sudo systemctl enable docker
```

Installing PMM Server

1. On your HammerDB client server, download the PMM Server docker image:

```
Docker pull percona/pmm-server:latest
```

2. Create PMM data container:

```
docker create \
-v /opt/prometheus/data \
-v /opt/consul-data \
-v /var/lib/mysql \
-v /var/lib/grafana \
--name pmm-data \
percona/pmm-server:latest /bin/true
```

3. Create PMM Server container by creating a systemd service unit file and adding the following:

```
[Service]
Restart=always
TimeoutStartSec=0
ExecStartPre=-/usr/bin/docker kill pmm-server
ExecStartPre=-/usr/bin/docker rm pmm-server
```

```
ExecStart=/usr/bin/docker run \  
--name pmm-server \  
--volume-from pmm-data \  
--name pmm-server \  
--restart always \  
-p=8080:80 \  
percona/pmm-server  
  
ExecStop=/usr/bin/docker kill pmm-server  
ExecStop=/usr/bin/docker rm pmm-server
```

4. Start and enable the services with systemctl command:

```
systemctl start pmm-server  
systemctl enable pmm-server
```

Installing PMM client on the server under test

1. Log into the Oracle database server as root.
2. Install the PMM Client package:

```
sudo yum install http://www.percona.com/downloads/percona-release/redhat/<LATEST_VERSION>  
sudo yum -y install pmm-client
```

3. Connect PMM Clients to the PMM Server:

```
pmm-admin config --server <IP address>:8080
```

4. Enable Linux performance metrics:

```
sudo pmm-admin add linux:metrics
```

Building the TPC-C-like schema

1. Log onto the HammerDB client as the Oracle user.
2. Change directories into the HammerDB directory:

```
cd HammerDB-3.3
```

3. Switch to the hammerdb command line interface:

```
./hammerdbcli
```

4. Change the following settings in the build script:

```
dbset db ora  
dbset bm tpc-c  
diset connection system_password <Password>  
diset connection instance ORCL  
diset tpcc count_ware 3750  
diset tpcc num_vu 16  
diset tpcc tpcc_def_tab tpcc  
diset tpcc tpcc_ol_tab tpcc_ol  
diset tpcc partition true  
diset tpcc hash_clusters true
```

5. Build the schema:

```
buildschema
```

6. After the build has finished, destroy the virtual users:

```
vudestroy
```

Running the TPC-C-like test

After building the schema and prior to running any tests, we took a full backup of the database with RMAN, which we used to restore the database back to its original build between runs.

1. Log onto the HammerDB client as the Oracle user.
2. Change directories into the HammerDB directory:

```
cd HammerDB-3.3
```

3. Switch to the hammerdb command line interface:

```
./hammerdbcli
```

4. Change the following settings in the load script:

```
dbset db ora
dbset bm tpc-c
diset connection system_password <Password>
diset connection instance ORCL
diset tpcc count_ware 3750
diset tpcc tpcc_def_tab tpcc
diset tpcc tpcc_ol_tab tpcc_ol
diset tpcc partition true
diset tpcc total_iterations 100000000
diset tpcc rampup 15
diset tpcc duration 30
diset tpcc ora_driver timed
loadscript
```

5. Create the virtual users:

```
vuset vu 384
vuset showoutput 1
vuset logtotemp 1
vuset unique 1
vucreate
```

6. Run the test:

```
vurun
```

7. When the test has completed, record the transactions per minute and delete the virtual users:

```
vudestroy
```

8. Restore the database, reboot each SUT and client server, and repeat steps 1 through 7 two more times for a total of three runs.

Oracle spfile

Oracle uses the spfile to set the database parameters at startup. We added several parameters to the default spfile to increase performance, and restarted the database with these settings.

```
orcl.__data_transfer_cache_size=0
orcl.__db_cache_size=240518168576
orcl.__inmemory_ext_roarea=0
orcl.__inmemory_ext_rwarea=0
orcl.__java_pool_size=0
orcl.__large_pool_size=2684354560
orcl.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
orcl.__pga_aggregate_target=120259084288
orcl.__sga_target=360777252864
orcl.__shared_io_pool_size=536870912
orcl.__shared_pool_size=33822867456
orcl.__streams_pool_size=0
orcl.__unified_pga_pool_size=0
*.disk_sector_size_override=TRUE
*.audit_file_dest='/u01/app/oracle/admin/orcl/adump'
*.audit_trail='none'
*.compatible='19.0.0'
*.control_files='+DATA/ORCL/CONTROLFILE/current.262.1040497183','+DATA/ORCL/CONTROLFILE/
current.261.1040497183'#Restore Controlfile
*.db_16k_cache_size=32G
*.db_4k_cache_size=32G
*.db_block_size=8192
*.db_create_file_dest='+DATA'
*.db_name='orcl'
*.db_recovery_file_dest='+DATA'
*.db_recovery_file_dest_size=500g
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=orclXDB)'
*.local_listener='LISTENER_ORCL'
*.nls_language='AMERICAN'
*.nls_territory='AMERICA'
*.open_cursors=2000
*.pga_aggregate_target=114688m
*.processes=7680
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=344064m
*.undo_tablespace='UNDOTBS1'
*.lock_sga=TRUE
*.use_large_pages='ONLY'
*.log_buffer=16G
*._enable_NUMA_interleave=TRUE
*._enable_NUMA_support=TRUE
*.fast_start_mttr_target=0
*.filesystemio_options='setall'
*.log_checkpoint_interval=0
*.log_checkpoint_timeout=0
*.log_checkpoints_to_alert=TRUE
*._in_memory_undo=TRUE
*._numa_shift_enabled=FALSE
*.disk_asynch_io=TRUE
*.undo_management='AUTO'
*.undo_retention=10
*._check_block_after_checksum=FALSE
*._collect_undo_stats=FALSE
*._db_block_check_objtyp=FALSE
*._db_block_prefetch_limit=0
*._db_writer_flush_imu=false
*._disable_highres_ticks=TRUE
*._disable_logging=FALSE
*._disable_selftune_checkpointing=TRUE
```



```
*._collect_undo_stats=FALSE
*._resource_manager_always_off=TRUE
*._trace_pool_size=0
*._undo_autotune=FALSE
*._use_adaptive_log_file_sync='FALSE'
*.aq_tm_processes=0
*._db_block_check_objtyp=FALSE
*._db_block_prefetch_limit=0
*._db_writer_flush_imu=false
*._disable_highres_ticks=TRUE
*._disable_logging=FALSE
*._disable_selftune_checkpointing=TRUE
*.plsql_code_type='NATIVE'
*.plsql_optimize_level=3
*.pre_page_sga=FALSE
*.query_rewrite_enabled='FALSE'
*.remote_login_passwordfile='EXCLUSIVE'
*.replication_dependency_tracking=FALSE
*.trace_enabled=FALSE
*.transactions_per_rollback_segment=1
*.commit_logging=BATCH
```

Read the report at <http://facts.pt/em5rf8b> ►

This project was commissioned by Dell Technologies.



Facts matter.®

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

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:

Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.