



The science behind the report:

Improving Oracle Database performance: Moving from older servers to 16th Generation Dell PowerEdge R960 servers with Intel Xeon Scalable processors

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 [Improving Oracle Database performance: Moving from older servers to 16th Generation Dell PowerEdge R960 servers with Intel Xeon Scalable processors](#).

We concluded our hands-on testing on July 26, 2023. 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 July 18, 2023 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to <http://facts.pt/calculating-and-highlighting-wins>. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Transactions per minute (TPM) results of our testing.

	Normalized TPM	Performance win for the Dell™ PowerEdge™ R960 solution (times as many TPM)
Dell PowerEdge R960 solution	7.9	-
Dell PowerEdge R940 solution	3.2	2.4
Dell PowerEdge R920 solution	1	7.9

Table 2: TPM results of our testing divided by rack units of each solution.

	Rack units (U)	Normalized TPM/U	Performance win for the Dell PowerEdge R960 solution (times as many TPM per total rack U)
Dell PowerEdge R960 solution	4	1.98	-
Dell PowerEdge R940 solution	3	1.07	1.8
Dell PowerEdge R920 solution	4	0.25	7.9

System configuration information

Table 3: Detailed information on the system we tested.

System configuration information	Dell PowerEdge R960
BIOS name and version	1.4.3
Operating system name and version/build number	Windows Server 2022 Datacenter Evaluation
Date of last OS updates/patches applied	5/5/2023
Power management policy	Performance
Processor	
Number of processors	4
Vendor and model	Intel® Xeon® Gold 6418H
Core count (per processor)	24
Core frequency (GHz)	2.10
Stepping	S3
Memory module(s)	
Total memory in system (TB)	2
Number of memory modules	32
Vendor and model	Hynix HMCG94AEBRA109N
Size (GB)	64
Type	DDR-5
Speed (MHz)	4,800
Speed running in the server (MHz)	4,800
Storage controller	
Vendor and model	1x Dell PERC H965i Front
Cache size (GB)	8
Firmware version	8.0.0.0.18-84
SAS SSD storage	
Number of drives	16
Drive vendor and model	Seagate ST1200MM0099
Drive size (TB)	1.1
Drive information	SAS SSD 12Gbps
NVMe SSD storage	
Number of drives	8
Drive vendor and model	Dell Ent NVMe® P5600
Drive size (TB)	1.6
Drive information	PCIe® NVMe SSD 16GT/s

System configuration information	Dell PowerEdge R960
Network adapter	
Vendor and model	Broadcom® NetXtreme E-Series
Number and type of ports	2x 25Gb Ethernet
Cooling fans	
Number	24
Power supplies	
Vendor and model	Dell CNDED002BH1T1M
Number of power supplies	4
Wattage of each (W)	2,400

How we tested

Testing overview

For this project, we tested the performance of a Dell PowerEdge R960 server using the TPROC-C workload from HammerDB. We compared the TPROC-C output to older HammerDB TPC-C-like results for previous-gen Dell PowerEdge R940 and PowerEdge R920 servers from one of our previous studies, which you can read here: <https://facts.pt/fhvcwjm>. In this testing, we used Oracle® pluggable databases (PDBs) in a bare metal environment, whereas in the previous study, we tested the previous-gen servers in virtual environments. We created a gold PDB, loaded it with the HammerDB TPROC-C dataset, and then cloned the PDB to the four PDBs we used for testing.

Using our methodology to aid your own deployments

While the methodology below describes in great detail how we accomplished our testing, it is not a deployment guide. However, because we include many basic installation steps for operating systems and testing tools, reading our methodology could help with your own installation.

Installing Red Hat Enterprise Linux 8.8

1. Boot to the Red Hat® Enterprise Linux® 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 on which to install the OS.
5. Under Other Storage Options, select I will configure partitioning, and click Done.
6. To create the partitions automatically, select Click here.
 - a. If the /home partition exists, remove it.
7. Expand the swap partition to 16GB.
8. Assign all remaining free space to the / partition.
9. Click Done.
10. Click Accept Changes.
11. Select Kdump.
12. Uncheck Enable kdump, and click Done.
13. Select Network & Hostname.
14. Enter the desired hostname for the system.
15. Turn on the desired network ports, and click Configure.
16. On the General tab, select Automatically connect to this network when it is available.
17. On the IPv4 Settings tab, choose the Method drop-down menu, and select Manual.
18. Under Addresses, click Add, and enter the desired static IP information for the server.
19. Enter the desired DNS information.
20. Click Save, and click Done.
21. Select Date & Time, and ensure the correct date, time, and time zone are set.
22. To add your NTP server, next to the Network Time On/Off switch, click the cog icon.
23. Add the IP address of your NTP server, and click +.
24. Uncheck all other NTP servers.
25. Click OK.
26. Click Done.
27. Click Software Selection.
28. Choose the Base Environment of Minimal Install.
29. Click Done.
30. Click Begin Installation.
31. Select Root Password.
32. Enter the desired root password, and click Done.
33. When the installation completes, select Reboot to restart the server.

Configuring Red Hat Enterprise Linux 8.8 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 Red Hat Enterprise Linux subscription:

```
subscription-manager register --username <Red Hat Enterprise Linux_USERNAME> --password <Red Hat Enterprise Linux_PASSWORD> --auto-attach
```

5. Install wget, sysstat, and gdisk:

```
yum install -y sysstat wget gdisk
```

6. Update Red Hat Enterprise Linux 8.8:

```
yum update -y
```

7. Disable auditd:

```
systemctl disable auditd
```

8. To disable transparent hugepages persistently, in /etc/default/grub, add transparent_hugepage=never to the end of the GRUB_CMDLINE_LINUX option, and run the following command to rebuild the grub.cfg file:

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

9. Reboot.
10. Log in as root.
11. Install the Oracle preinstall:

```
dnf install oracle-database-preinstall-19c
```

12. Create groups for Oracle Automatic Storage Management (ASM):

```
groupadd -g 54327 asmdba
groupadd -g 54328 asmoper
groupadd -g 54329 asmadmin
```

13. To add the ASM groups, modify the Oracle user:

```
usermod -G asmdba,asmoper,asmadmin -a oracle
```

14. Create passwords for the Oracle account with passwd:

```
passwd oracle
```

15. On a separate computer, navigate to the Red Hat Enterprise Linux Labs page for the Red Hat Enterprise Linux Tuner for Oracle (https://access.redhat.com/labs/rhel_tfo/), and fill out the proper details for your system. (We noted the SGA as 384 GB.) Click the button to generate the script for your server:

```
Red Hat Enterprise Linux Version: Red Hat Enterprise Linux 8
Oracle Group Name/ID: 54321
System Global Area (SGA): 384 GB
Maximum number of open file descriptors: 65536 (Default)
```

16. Copy the Red Hat Enterprise Linux Tuner script to your server.
17. To make Red Hat Enterprise Linux Tuner executable, change the permissions:

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

18. On the server, accept all changes, and execute the Red Hat Enterprise Linux Tuner:

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

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

```
mkdir -p /u01/app/19.3.0.0/grid
mkdir -p /u01/app/oracle/product/19.3.0.0/dbhome_1
chown -R oracle:oinstall /u01
chmod -R 775 /u01/
```

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

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

21. Log into the Oracle user:

```
su - oracle
```

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

```
export TMP=/tmp
export TMPDIR=$TMP
export ORACLE_HOSTNAME=<HOSTNAME>
export ORACLE_UNQNAME=orcl
export ORACLE_BASE=/u01/app/oracle
export GRID_HOME=/u01/app/19.3.0.0/grid
export DB_HOME=$ORACLE_BASE/product/19.3.0.0/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'
```

23. 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
```

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

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

25. Download the OracleASM support and lib packages from here:

- <https://www.oracle.com/linux/downloads/linux-asmlib-v8-downloads.html>
- https://public-yum.oracle.com/repo/OracleLinux/OL8/addons/x86_64/index.html

26. Copy the packages to your Red Hat Enterprise Linux server.

27. From the directory to which you copied the packages, install the Oracle ASM lib and support packages:

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

28. 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
```

29. Run the following command to initialize oracleasm:

```
oracleasm init
```

30. Create a partition on all disks using gdisk.

31. Run the following commands to configure all 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 DATA5 /dev/<DISK5>
oracleasm createdisk DATA6 /dev/<DISK6>
oracleasm createdisk LOG1 /dev/<DISK7>
oracleasm createdisk LOG2 /dev/<DISK8>
```

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 External.
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, enter 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. In the Prerequisite Checks screen, make sure there are no errors.
17. In the Summary screen, verify that everything is correct, and click Finish to install Oracle Grid Infrastructure.
18. When 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 External 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. When the GUI installer loads, on the Select Configuration Option, select Set Up Software Only, and click Next.
6. On the Select Database Installation Option, select Single instance database installation, and click Next.
7. On Select Database Edition, select Enterprise Edition, and click Next.
8. For Specify Installation Location, leave the default Oracle base, and click Next.

9. For the Privileged Operating System groups, keep the default, and click Next.
10. On the Perform Prerequisite Checks, wait for the verification results, fix all errors, and click Next.
11. Review the summary, and click Install.
12. 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
```

13. Return to the prompt, and click OK.
14. 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. To open the Database configuration assistant, type `dbca`, and 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 Data Warehouse, and click Next.
7. Enter a Global database name and the appropriate SID, and uncheck Create as Container database, and 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, select Specify Fast Recovery Area.
11. In the drop-down menu, select ASM, select +DATA for the Fast Recovery Area, and for the size, enter 500GB
12. At the Specify Network Configuration Details, 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, 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

1. Alter the tablespaces on both systems as shown below.
2. Type `sqlplus / as sysdba` to enter SQL prompt, and type the following:

```
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 200G
BLOCKSIZE 4K;
ALTER DATABASE ADD LOGFILE GROUP 2 ( '+LOG/redo02.log' ) SIZE 200G
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 PLUGGABLE DATABASE DS00 ADMIN USER <USER> IDENTIFIED BY <PASSWORD>;
ALTER PLUGGABLE DATABASE DS00 OPEN;
ALTER SESSION SET CONTAINER=DS00;

CREATE BIGFILE TABLESPACE "TPCC"
DATAFILE '+DATA/ORCL/<PDB DIRECTORY>/DATAFILE/tpcc.dbf' SIZE 40G AUTOEXTEND ON NEXT 1G
BLOCKSIZE 8K
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;

CREATE BIGFILE TABLESPACE "TPCC_OL"
DATAFILE '+DATA/ORCL/<PDB DIRECTORY>/DATAFILE/tpcc_ol.dbf' SIZE 20G AUTOEXTEND ON NEXT 1G
BLOCKSIZE 16K
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;

ALTER DATABASE DATAFILE '+DATA/ORCL/<PDB DIRECTORY>/DATAFILE/<UNDO FILE>' RESIZE 32760M;

```

Creating the HammerDB client VMs

Our infrastructure server arrived preinstalled with Windows Server 2022 Datacenter. We created our HammerDB client VMs using Windows Hyper-V.

1. Boot to the Red Hat Enterprise Linux 8.8 installation media.
2. Select Install Red Hat Enterprise Linux 8.8.
3. Select English, and click Continue.
4. For the Installation destination, select automatic partitioning, and click Done.
5. For the Software Selection, select Minimal Install, and click Done.
6. Configure a static IP address, and click Done.
7. Set a password for the root user account, and click Done.
8. Click Begin Installation.

Configuring the client VM for Oracle and HammerDB

1. Log in as the root user.
2. Disable the firewall:

```

systemctl stop firewalld
systemctl disable firewalld

```

3. Attach the Red Hat Enterprise Linux subscription:

```

subscription-manager register --user-name [Username] --password [Password] --auto-attach

```

4. Disable SELinux:

```

vi /etc/selinux/config
SELINUX=disabled

```

5. Install sysstat, gcc, gcc-c++:

```
yum install -y sysstat gcc gcc-c++
```

6. Install EPEL repo:

```
yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

7. Update Red Hat Enterprise Linux 8.8:

```
yum update -y
```

8. Disable auditd:

```
systemctl disable auditd
```

9. Reboot.

10. Install compat-libstdc++-33:

```
yum install -y compat-libstdc++-33
```

11. Install Oracle Database Preinstall:

```
yum install oracle-database-preinstall-19c
```

12. Create a password for the Oracle user with passwd:

```
passwd oracle
```

13. Log into the Oracle user with X11 forwarding:

```
ssh -Y oracle@<IP_ADDRESS>
```

14. Download the Oracle Database 19c client zip file, and move it to the Oracle home folder.

15. Unzip LINUX.X64_193000_client.zip:

```
unzip -q LINUX.X64_193000_client.zip
```

16. Change directories into the unzipped folder:

```
cd client/
```

17. Run the client installer:

```
./runInstaller
```

18. At the Select Installation Type screen, select Administrator, and click Next.
19. Leave the defaults for the install location, and click Next.
20. Leave the defaults for the inventory directory and group name, and click Next.
21. At the prerequisite check screen, fix any prerequisite errors, and continue.
22. Click Install.

23. Run the root script as the root user when prompted:

```
/home/oracle/app/oraInventory/orainstRoot.sh
```

24. Click OK.
25. Click Close.
26. Download version 4.7 of HammerDB.
27. Install HammerDB:

```
tar -xzf HammerDB-4.7-RHEL8.tar.gz
```

28. Add the following lines to the `.bash_profile` to point HammerDB to the Oracle libraries:

```
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
```

29. Change directories to the tns admin directory:

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

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

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

31. 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)))
ORCL=(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=DS00))(ADDRESS=(PROTOCOL=tcp)(HOST=[IP_
ADDRESS])(PORT=1521)))
ORCL=(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=DS01))(ADDRESS=(PROTOCOL=tcp)(HOST=[IP_
ADDRESS])(PORT=1521)))
ORCL=(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=DS02))(ADDRESS=(PROTOCOL=tcp)(HOST=[IP_
ADDRESS])(PORT=1521)))
ORCL=(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=DS03))(ADDRESS=(PROTOCOL=tcp)(HOST=[IP_
ADDRESS])(PORT=1521)))
ORCL=(DESCRIPTION=(CONNECT_DATA=(SERVICE_NAME=DS04))(ADDRESS=(PROTOCOL=tcp)(HOST=[IP_
ADDRESS])(PORT=1521)))
```

32. Test the connection to make sure you can ping the Oracle server with the ORCL and DS00 service names:

```
tnsping ORCL
tnsping DS00
```

Building the TPROC-C schema

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

```
cd HammerDB-4.7
```

3. Switch to the HammerDB command line interface:

```
./hammerdbcli
```

4. Change the following steps in the build script:

```
dbset db ora
dbset bm tproc-c
diset connection system_user system
diset connection system_password [Password]
diset connection instance ds00
diset tpcc count_ware 400
diset tpcc num_vu 40
diset tpcc tpcc_user tpcc
diset tpcc tpcc_pass [Password]
diset tpcc tpcc_def_tab tpcc
diset tpcc tpcc_ol_tab tpcc_ol
diset tpcc tpcc_def_temp temp
diset tpcc partition true
diset tpcc hash_clusters true
```

5. Build the schema:

```
buildschema
```

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

```
vudestroy
```

Running the TPROC-C test

After we built the TPROC-C schema on our client VM, we powered it off and cloned out three additional client VMs, and then changed the hostname and IP of each.

1. Log into the system under test as the Oracle user.
2. Switch to the database environment:

```
db_env
```

3. Log into the Oracle DB:

```
sqlplus / as sysdba
```

4. Shut down the database:

```
shutdown;
```

5. Start the database using the pfile we included in the [Scripts we used for testing](#) section:

```
startup pfile='/home/oracle/pfile.ora';
```

6. Close the DS00 pluggable database:

```
alter pluggable database ds00 close;
```

7. Clone out the four PDBs used for testing:

```
create pluggable database ds01 from ds00;  
create pluggable database ds02 from ds00;  
create pluggable database ds03 from ds00;  
create pluggable database ds04 from ds00;
```

8. Open the PDBs under test:

```
alter pluggable database all except ds00 open;
```

9. Log into each HammerDB client VM as the Oracle user.
10. Change directories into the HammerDB directory:

```
cd HammerDB-4.7
```

11. Run the test using the HammerDB run script included in the [Scripts we used for testing](#) section on each client:

```
./hammerdbcli auto run.tcl
```

12. When the test completes, log into the SUT as the Oracle user.
13. Close the pluggable databases under test:

```
alter pluggable database all close;
```

14. Drop the pluggable databases used for testing:

```
drop pluggable database ds01 including datafiles;  
drop pluggable database ds02 including datafiles;  
drop pluggable database ds03 including datafiles;  
drop pluggable database ds04 including datafiles;
```

15. Reboot the system under test.
16. Complete steps 1 through 13 two more times for a total of three test runs. We report the median of the three runs.

Scripts we used for testing

pfile.ora

```
orcl1.__data_transfer_cache_size=0
orcl1.__db_cache_size=1145682526208
orcl1.__inmemory_ext_roarea=0
orcl1.__inmemory_ext_rwarea=0
orcl1.__java_pool_size=0
orcl1.__large_pool_size=1073741824
orcl1.__oracle_base='/u01/app/oracle'#ORACLE_BASE set from environment
orcl1.__pga_aggregate_target=8589934592
orcl1.__sga_target=1298153865216
orcl1.__shared_io_pool_size=1073741824
orcl1.__shared_pool_size=142807662592
orcl1.__streams_pool_size=0
orcl1.__unified_pga_pool_size=0
*.disk_sector_size_override=TRUE
*.audit_file_dest='/u01/app/oracle/admin/orcl/adump'
*.audit_trail='db'
*.commit_logging='IMMEDIATE'
*.commit_wait='NOWAIT'
*.compatible='19.0.0'
*.control_files='+DATA/ORCL/CONTROLFILE/current.275.1138181487','+DATA/ORCL/CONTROLFILE/
current.274.1138181487'
*.db_16k_cache_size=4294967296
*.db_block_checking='false'
*.db_block_checksum='false'
*.db_block_size=8192
*.db_cache_size=34359738368
*.db_create_file_dest='+DATA'
*.db_file_multiblock_read_count=4
*.db_name='orcl'
*.db_recovery_file_dest='+DATA'
*.db_recovery_file_dest_size=16g
*.db_writer_processes=4
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers=(PROTOCOL=TCP) (SERVICE=orclXDB)'
*.dml_locks=500
*.enable_pluggable_database=true
*.local_listener='LISTENER_ORCL'
*.lock_sga=TRUE
*.log_buffer=1073709056
*.log_checkpoint_timeout=0
*.nls_language='AMERICAN'
*.nls_territory='AMERICA'
*.open_cursors=2000
*.parallel_max_servers=0
*.parallel_min_servers=0
*.pga_aggregate_target=8589934592
*.plsql_code_type='NATIVE'
*.plsql_optimize_level=3
*.pre_page_sga=FALSE
*.processes=15360
*.query_rewrite_enabled='FALSE'
*.remote_login_passwordfile='EXCLUSIVE'
*.replication_dependency_tracking=FALSE
*.result_cache_max_size=0
*.sga_target=1237928m
*.star_transformation_enabled='TRUE'
*.trace_enabled=FALSE
*.undo_retention=2
*.undo_tablespace='UNDOTBS1'
*.use_large_pages='auto_only'
```

run.tcl

In this script we changed one parameter (instance) depending on the HammerDB client VM it was being started from. Each VM should correspond with the PDB, i.e. VM01's instance would be PDB DS01, VM02's instance would be PDB DS02, etc.

```
#!/bin/tclsh
puts "SETTING CONFIGURATION"
global complete
proc wait_to_complete {} {
  global complete
  set complete [vucomplete]
  if (!$complete) { after 5000 wait_to_complete } else { exit }
}

dbset db ora

diset connection system_user system
diset connection system_password [Password]
diset connection instance ds[01-04]
diset tpcc count_ware 400
diset tpcc tpcc_user tpcc
diset tpcc tpcc_pass [Password]
diset tpcc tpcc_def_tab tpcc
diset tpcc tpcc_ol_tab tpcc_ol
diset tpcc tpcc_def_temp temp
diset tpcc partition true
diset tpcc hash_clusters true
diset tpcc total_iterations 100000000000
diset tpcc keyandthink false
diset tpcc checkpoint false
diset tpcc ora_driver timed
diset tpcc rampup 10
diset tpcc duration 20
diset tpcc ora_timeprofile true

vuset logtotemp 1

loadscript

vuset vu 128
vuset showoutput 1
vucreate

vurun

wait_to_complete
vwait forever
```

Read the report at <https://facts.pt/Gc6b1TG>

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.