Behind the Report:
Can Dell™ PowerEdge™ R450 and R550 Servers Meet the Flexibility and Performance Needs of SMBs?

This document provides the system-configuration details and step-by-step procedures that Prowess Consulting used to perform benchmark testing on three Dell Technologies™ platforms:

- Dell™ PowerEdge™ R440
- Dell PowerEdge R450
- Dell PowerEdge R540
- Dell PowerEdge R550

For the full analysis, read the report, "Can Dell PowerEdge R450 and R550 Servers Meet the Flexibility and Performance Needs of SMBs?"

Testing was concluded on August 8, 2022.

Server Configurations

	Dell™ PowerEdge™ R440	Dell™ PowerEdge™ R450	Dell™ PowerEdge™ R540	Dell™ PowerEdge™ R550
Hardware				
Processor	Intel® Xeon® Silver 4214	Intel® Xeon® Silver 4314	Intel® Xeon® Gold 5217	Intel® Xeon® Gold 5317
Number of CPUs	2	2	2	2
Cores per CPU	12	16	8	12
Cores/threads total	24/48	32/64	16/32	24/48
Frequency (base/SCT/MCT)	2.20 GHz	2.40 GHz	3.00 GHz	3.00 GHz
Storage controller 01	Dell™ PowerEdge RAID Controller (PERC) H740P Adapter	Dell™ PowerEdge RAID Controller (PERC) H745 Front Adapter	Dell™ PowerEdge RAID Controller (PERC) H740P Adapter	Dell™ PowerEdge RAID Controller (PERC) H755 Front Adapter
Disk	960 GB Micron® MTFDDAK960TDS	960 GB Micron® MTFDDAK960TDS	960 GB WD® WUSTVA196BSS200	960 GB WD® WUSTVA196BSS200
Number of disks	4	4	6	6
Installed memory	128 GB	256 GB	256 GB	256 GB
Memory DIMM	Micron 16 GB	Hynix 16 GB	Micron 32 GB	Micron 32 GB
Memory speed	2,400 megatransfers per second (MT/s)	2,666 MT/s	2,666 MT/s	2,933 MT/s
Number of memory DIMMs	8	8	8	8

BIOS version	2.14.2	1.6.5	2.14.2	1.6.5
Operating system (OS) performance profile	Oracle	Oracle	Oracle	Oracle
Software				
OS	Red Hat® Enterprise Linux® 8.6 (Ootpa)	Red Hat® Enterprise Linux® 8.6 (Ootpa)	Red Hat® Enterprise Linux® 8.6 (Ootpa)	Red Hat® Enterprise Linux® 8.6 (Ootpa)
OS kernel	Linux® 4.18.0-372.16.1.el8_6.x86_64	Linux® 4.18.0-372.16.1.el8_6.x86_64	Linux® 4.18.0-372.16.1.el8_6.x86_64	Linux® 4.18.0-372.19.1.el8_6.x86_64
Database	MySQL® Community Server	MySQL® Community Server	MySQL® Community Server	MySQL® Community Server
Benchmarking tools				
Database performance	HammerDB 4.4	HammerDB 4.4	HammerDB 4.4	HammerDB 4.4
Virtual machine (VM) performance	Stress-NG	Stress-NG	Stress-NG	Stress-NG

Testing Procedures

We completed the following testing procedures on the Dell PowerEdge R440, PowerEdge R450, PowerEdge R540, and PowerEdge R550 servers.

MySQL® Community Server

We completed the following steps to test MySQL® Community Server:

1. Create a new virtual disk:
    a. **Name: Disk**
    b. **RAID: RAID-5**
    c. **Read Policy: No read ahead**
    d. **Write policy: Write through**
2. Click **Next**, and then click Next to confirm.
3. Click **Add to pending**.
4. Click **Apply Now** to create the disk.
5. Click **Apply Now**.
6. Connect to Dell™ Lifecycle Controller.
7. Select **Get the latest firmware**.
8. Select **Dell Website**, and then click **Next**.
9. Apply all updates.
10. Boot the server into **Red Hat Enterprise Linux 8.6 Installation**.
    a. At the **Welcome to Red Hat Enterprise Linux 8.6** screen, select English, and then click Continue.
    b. Click **Time & Date**, and then select the testing time zone.

c. Select **Root Password**:

   i. Enter a root password.

   ii. Enter to confirm the root password.

   iii. Click **Done**.

d. Click **User Creation**:

   i. Enter the following details:

      1. **Full name**

      2. **User name**

      3. **Password**

      4. **Confirm password**

   ii. Check to make this user is an administrator.

   iii. Click **Done**.

e. Click **Software Selection**:

   i. Select **Server**.

   ii. Click **Done**.

f. Click **Installation Destination**:

   i. Select the disk.

   ii. Select **Custom**.

   iii. Click **Done**.

   iv. Under **Manual Partitioning**, click **Create here** to create them automatically.

   v. Adjust the partitions as follows:

      1. **Home: 100 GiB**

      2. **Swap: 16 GiB**

      3. **Root**: Apply the remainder of available storage

   vi. Click **Done**.

   vii. At the **Summary of changes** screen, click **Accept Changes**.

g. Click **Network & Host Name**:

   i. Update the host name.

   ii. Click **Done**.

h. Click **Begin Installation** to install Red Hat Enterprise Linux.

i. At the **Installation Progress** screen, click **Reboot System** once the operating system is installed.

11. After the system is rebooted, log in as the root or user.

12. Run the following command to get the IP address:

```
ip a
```

13. From the client system, enter the following command to create a Secure Shell (SSH) connection into the server:

```
ssh root@server IP
```

14. Run the following command to install updates:

```
dnf update -y && dnf upgrade -y
```

15. Run the following command to install fio:

```
dnf install fio
```

16. Run the following command to precondition the disks:

```
fio --ioengine=libaio --invalidate=1 --fsync_on_close=1 --direct=1 --output=steady.
tmp --norandommap --numjobs=8 --name=/dev/sda --rw=write --bs=4k --iodepth=32
--randrepeat=0 --size=2000G
```

17. Run the following command to disable the default MySQL module:

```
sudo dnf module disable mysql
```

18. Run the following commands to install MySQL Community Server:

```
setenforce 0
sed -i 's/SELINUX=.*/SELINUX=Permissive/' /etc/selinux/config
systemctl disable --now firewalld
dnf install -y tuned-profiles-oracle
tuned-adm profile oracle
tuned-adm active
sed -i -e '/vm.swappiness/d' -e '/fs.aio-max-nr/d' /etc/sysctl.conf
cat <<EOF >>/etc/sysctl.conf
vm.swappiness = 1
fs.aio-max-nr = 1048576
EOF
sysctl -p
sudo dnf module disable mysql
subscription-manager repos --enable codeready-builder-for-rhel-8-$(arch)-rpms
dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
dnf install -y epel-release
dnf install -y wget vim tar zip unzip lz4 pigz nmon sysstat numactl atop dstat
dnf install -y https://dev.mysql.com/get/mysql-community-client-plugins-8.0.29-1.el8.
x86_64.rpm
dnf install -y https://dev.mysql.com/get/mysql-community-common-8.0.29-1.el8.x86_64.rpm
dnf install -y https://dev.mysql.com/get/mysql-community-libs-8.0.29-1.el8.x86_64.rpm
dnf install -y https://dev.mysql.com/get/mysql-community-client-8.0.29-1.el8.x86_64.rpm
dnf install -y https://dev.mysql.com/get/mysql-community-icu-data-files-8.0.29-1.el8.
x86_64.rpm
dnf install -y https://dev.mysql.com/get/mysql-community-server-8.0.29-1.el8.x86_64.rpm
dnf --disablerepo=AppStream install -y mysql-community-server
systemctl disable --now mysqld
```

19. Run the following command to back up the **my.cnf** file:

```
cp /etc/my.cnf /etc/my.cnf.bkup
```

20. Run the following command to edit the my.cnf file:

```
nano my.cnf
```

21. Add the following settings:

```
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/8.0/en/server-configuration-defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove the leading "# " to disable binary logging
# Binary logging captures changes between backups and is enabled by
# default. It's default setting is log_bin=binlog
# disable_log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
#
# Remove leading # to revert to previous value for default_authentication_plugin,
default-authentication-plugin=mysql_native_password
datadir=/mysql/data
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
port=3306
bind_address=0.0.0.0
max_connections=4000
table_open_cache=8000
table_open_cache_instances=16
back_log=1500
default_password_lifetime=0
ssl=0
performance_schema=OFF
max_prepared_stmt_count=128000
skip_log_bin=1
character_set_server=latin1
collation_server=latin1_swedish_ci
transaction_isolation=REPEATABLE-READ
innodb_file_per_table
innodb_log_file_size=1024M
innodb_log_files_in_group=8 #scale
innodb_open_files=4000
innodb_buffer_pool_size=24000M #scale
innodb_buffer_pool_instances=16
innodb_log_buffer_size=64M
innodb_doublewrite=0
```

```
innodb_thread_concurrency=0
innodb_flush_log_at_trx_commit=0
innodb_max_dirty_pages_pct=90
innodb_max_dirty_pages_pct_lwm=10
join_buffer_size=32K
sort_buffer_size=32K
innodb_use_native_aio=1
innodb_stats_persistent=1
innodb_spin_wait_delay=6
innodb_max_purge_lag_delay=300000
innodb_max_purge_lag=0
innodb_flush_method=O_DIRECT_NO_FSYNC
innodb_checksum_algorithm=none
innodb_io_capacity=1000
innodb_io_capacity_max=2000
innodb_lru_scan_depth=9000
innodb_change_buffering=none
innodb_read_only=0
innodb_page_cleaners=4
innodb_undo_log_truncate=off
innodb_adaptive_flushing=1
innodb_flush_neighbors=0
innodb_read_io_threads=16
innodb_write_io_threads=16
innodb_purge_threads=4
innodb_adaptive_hash_index=0
innodb_monitor_enable='%'
```

22. Run the following command to edit the my.cnf file:

```
systemctl start mysqld
```

23. Run the following command to edit the my.cnf file:

```
systemctl status mysqld
```

24. Run the following command to install Dstat and atop:

```
dnf install atop dstat
```

25. Run the following command to download HammerDB:

```
curl -kLJO https://github.com/TPC-Council/HammerDB/releases/download/v4.4/HammerDB-4.4-Linux.tar.gz
```

26. Run the following command to extract HammerDB:

```
tar -xf HammerDB-4.4-Linux.tar.gz
```

27. Run the following command to retrieve a temp password for the MySQL root user:

```
sudo grep 'temporary password' /var/log/mysqld.log
```

28. Run the following command to log in to MySQL with the temporary password:

```
mysql -u root -p
```

29. Run the following command to set the root password:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'P@$$w0rd1';
```

30. Run the following commands to create a test user:

```
CREATE USER 'test'@'localhost' IDENTIFIED BY 'TestPassword123!';
GRANT ALL PRIVILEGES ON *.* TO 'test'@'localhost'
WITH GRANT OPTION;
CREATE USER 'test'@'%' IDENTIFIED BY 'TestPassword123!';
GRANT ALL PRIVILEGES ON *.* TO 'test'@'%'
WITH GRANT OPTION'
```

31. Run the following command to update the HammerDB mysql.xml configuration file:

```
nano /root/HammerDB-4.4/config/mysql.xml
```

32. Update the following areas:

```
<mysql>
       <connection>
       <mysql_host>127.0.0.1</mysql_host>
       <mysql_port>3306</mysql_port>
       <mysql_socket>/var/lib/mysql/mysql.sock</mysql_socket>
       </connection>
       <tpcc>
       <schema>
              <mysql_count_ware>1</mysql_count_ware>
              <mysql_num_vu>1</mysql_num_vu>
              <mysql_user>test</mysql_user>
              <mysql_pass>TestPa$$w0rd123</mysql_pass>
              <mysql_dbase>tpcc</mysql_dbase>
              <mysql_storage_engine>innodb</mysql_storage_engine>
              <mysql_partition>false</mysql_partition>
              <mysql_prepared>false</mysql_prepared>
       </schema>
       <driver>
              <mysql_total_iterations>10000000</mysql_total_iterations>
              <mysql_raiseerror>false</mysql_raiseerror>
              <mysql_keyandthink>false</mysql_keyandthink>
              <mysql_driver>test</mysql_driver>
              <mysql_rampup>5</mysql_rampup>
              <mysql_duration>20</mysql_duration>
              <mysql_allwarehouse>true</mysql_allwarehouse>
              <mysql_timeprofile>true</mysql_timeprofile>
              <mysql_async_scale>false</mysql_async_scale>
              <mysql_async_client>10</mysql_async_client>
              <mysql_async_verbose>false</mysql_async_verbose>
              <mysql_async_delay>1000</mysql_async_delay>
              <mysql_connect_pool>false</mysql_connect_pool>
       </driver>
       </tpcc>
```

33. Run the following command to start the HammerDB command-line interface (CLI):

```
./hammerdbcli
```

34. Run the following command to set MySQL as the database engine:

```
dbset db mysql
```

35. Run the following command to set the benchmark:

```
dbset bm TPROC-C
```

36. Run the following command to set the HammerDB warehouse count:

```
diset tpcc mysql_count_ware 500
OR
diset tpcc mysql_count_ware 1000
```

37. Run the following command to set the number of virtual users to create the HammerDB test database:

```
diset tpcc mysql_num_vu 50
```

38. Run the following command to build the database schema:

```
buildschema
```

39. Enter the following command to back up the database:

```
mysqldump -u root -p tpcc > /mysql/data/tpcc500_backup.sql
OR
mysqldump -u root -p tpcc > /mysql/data/tpcc1000_backup.sql
```

40. Run the following command to create a mysql.tcl test script:

```
nano mysql.tcl
```

41. Copy the following into the script and save:

```
#!/bin/tclsh
proc runtimer { seconds } {
set x 0
set timerstop 0
while {!$timerstop} {
incr x
after 2000
       if { ![ expr {$x % 60} ] } {
              set y [ expr $x / 60 ]
              puts "Timer: $y minutes elapsed"
       }
update
if {  [ vucomplete ] || $x eq $seconds } { set timerstop 1 }
       }
return
}
puts "MySQL Test Started"
dbset db mysql
dbset bm TPC-C
vuset logtotemp 1
vuset unique 1
loadscript
foreach z {5 10 20 50} {
puts "$z VU test"
```

```
vuset vu $z
vucreate
vurun
runtimer 1500
vudestroy
after 1510
}
puts "MySQL Test Complete"
```

42. Run the following command to enter into the HammerDB CLI:

```
./hammdbcli
```

43. For testing purposes, run the following command three times, taking the median of the test output values for a score:

```
source mysql.tcl
```

44. Collect metrics during the test time using atop, Dstat, and nmon.

Red Hat® Enterprise Linux® Virtualization Load-Capacity Testing

The following steps were taken to test VM load capacity:

1. Create a new virtual disk:
    a. **Name: Disk**
    b. **RAID: RAID-5**
    c. **Read Policy: No read ahead**
    d. **Write policy: Write through**
2. Click **Next**, and then click **Next** to confirm.
3. Click **Add to pending**.
4. Click **Apply Now** to create the disk.
5. Create the virtual disk, and then click **Add to pending**.
6. Click **Apply Now**.
7. Connect to Dell Lifecycle Controller.
8. Select **Get the latest firmware**.
9. Select **Dell Website**, and then click **Next**.
10. Apply all updates.
11. Boot the server into **Red Hat Enterprise Linux 8.6 Installation**.
12. At the **Welcome to Red Hat Enterprise Linux 8.6** screen, select **English**, and then click **Continue**.
13. Click **Time & Date,** and then select the testing time zone.

14. Select **Root Password**:
    a. Enter a root password.
    b. Enter to confirm the root password.
    c. Click **Done**.
15. Click **User Creation**:
    a. Enter the following details:
        i. **Full name**
        ii. **User name**
        iii. **Password**
        iv. **Confirm password**
    b. Check to make this user an administrator.
    c. Click **Done**.
16. Click Software Selection:
    a. Select **Server**.
    b. Click **Done**.
17. 17. Click **Installation Destination**:
    a. Select the disk.
    b. Select **Custom**.
    c. Click **Done**.
    d. Under **Manual Partitioning**, click **Create here to create them automatically**.
    e. Adjust the partitions as follows:
        i. **Home: 100 GiB**
        ii. **Swap: 16 GiB**
        iii. **Root**: Apply the remainder of available storage
    f. Click **Done**.
    g. At the **Summary of changes** screen, click **Accept Changes**.
18. Click **Network & Host Name**:
    a. Update the host name.
    b. Click **Done**.

19. Click **Begin Installation** to install Red Hat Enterprise Linux.
20. At the **Installation Progress** screen, click **Reboot System** once the operating system is installed.
21. After the system is booted, log in as the root or user.
22. Enter the following command to get the IP address:

```
ip a
```

23. From the client system, enter the following command to create a SSH connection into the server:

```
ssh root@server ip
```

24. Run the following command to install updates:

```
dnf update -y && dnf upgraded -y
```

25. Run the following command to format the data disk:

```
mkfs.xfs -f /dev/sda
```

26. Run the following command to add the mount point to fstab:

```
echo '/dev/sda /virtualmachines/vms xfs defaults 0 0' >> /etc/fstab
```

27. Run the following commands to disable security policies for the purposes of testing:

```
setenforce 0
sed -i 's/SELINUX=.*/SELINUX=Permissive/' /etc/selinux/config
systemctl disable --now firewalld
```

28. Run the following command to install Red Hat virtualization:

```
yum module install virt
```

29. Run the following command to install additional virtualization-management packages:

```
yum install virt-install virt-viewer
```

30. Run the following command to start the virtualization service:

```
systemctl start libvirtd
```

31. Run the following command to enable VM management in the web console:

```
yum install cockpit-machines
```

32. Run the following command to enable the Red Hat cockpit:

```
sudo systemctl enable --now cockpit.socket
```

33. Run the following command to check the status for virtualization:

```
virt-host-validate
```

34. If prompted, run the following command to enable intel_iommu support:

```
grubby --update-kernel=ALL --args="intel_iommu=on"
```

35. Run the following command to download the CentOS 8 iso:

```
curl -kLJO http://mirror.mia.velocihost.net/centos/8-stream/isos/x86_64/CentOS-Stream-8-x86_64-20220719-dvd1.iso
```

36. Open a web browser to the systems IP: **https://<*ip*>:9090**.

    a. Select **Virtual Machines**.

    b. Select **Storage Pools**.

    c. Click **Create Storage Pools**.

    d. Enter the following details:

        i. **Name**: name

        ii. **Type**: Filesystem directory

        iii. **Target path: /virtualmachines**

        iv. **Startup: Start pool when hosts boots**

    e. Click **Create**.

    f. Click **Activate**.

    g. Click **Storage Volumes**.

    h. Click **Create Volume**.

    i. Enter the following details:

        i. **Name**: vm name

        ii. **Size: 40 GiB**

        iii. **Format: qcow2**

    j. Click **Virtual Machines**:

        i. Select **Create VM**.

        ii. Enter the following details:

            1. **Name: VM1**

            2. **Installation Type: Local install media**

            3. **Installation Source**: <location of downloaded iso>

            4. **Operating system: CentOS Stream 8**

            5. **Storage**: <storage name>

            6. **Volume**: <previously created volume>

            7. **Memory: 8 GiB**

iii. Click **Create**.

iv. Click **Install**.

k. Continue with the VM installation.

l. At the **Welcome to CentOS** page, click **Continue**.

m. **Root Password** page, enter a password and confirm.

n. Create a user:

i. **Full name**: <user>

ii. **User name**: <user>

iii. **Password**: <password>

iv. **Confirm password**: <password>

o. Click **Done**.

p. Click **Installation Destination**, and then click **Done**.

q. Click **Begin Installation**.

r. When installation is completed, click **Reboot System**.

37. Log in to the CentOS VM.

38. Enter the following command to update crontab:

```
sudo crontab -e
```

39. Enter the following information:

```
*/5 * * * * /opt/scripts/startup.sh
```

40. Run the following command to create a scripts director in /opt:

```
mkdir -p /opt/scripts
```

41. Run the following command to create startup.sh:

```
nano /opt/script/startup.sh
```

42. Copy the following to startup.sh:

```
#! /bin/bash
stress-ng --cpu 2 --io 2 --vm 4 --vm-bytes 1G --timeout 5m
```

43. Run the following command to install stress-ng; this tool is used to simulate a load on the VM:

```
dnf install stress-ng -y
```

44. Copy the VM up to 100 times across all servers.

45. Start the first five VMs and then one VM every 60 seconds until resources are fully allocated.
46. Collect metrics during the test time using atop, Dstat, and nmon.