Behind the Report—Testing Addendum:

Combined Testing for Dell™ PowerEdge™ and Supermicro® Servers

This document provides the system-configuration details and step-by-step procedures that Prowess Consulting engineers used to test and compare the deployment and performance of the following servers:

- Dell™ PowerEdge™ HS5610
- Dell™ PowerEdge™ HS5620
- Supermicro® SuperServer® SYS-120C-TN10R
- Supermicro® SuperServer® SYS-620C-TN12R

For the full analysis, read the report.

Testing was concluded on August 14, 2023.

Server Configurations

	Dell™ PowerEdge™ HS5610	Dell™ PowerEdge™ HS5620	Supermicro® SuperServer® SYS-120C-TN10R	Supermicro® SuperServer® SYS-620C-TN12R
CPU	Intel® Xeon® Gold 6448Y	Intel® Xeon® Gold 6448Y	Intel® Xeon® Gold 6448Y	Intel® Xeon® Gold 6448Y
Number of CPUs	2	2	2	2
Cores/Threads Per CPU	32/64	32/64	32/64	32/64
Cores/Threads Total	64/128	64/128	64/128	64/128
Frequency	2.10 MHz	2.10 MHz	2.10 MHz	2.10 MHz
Storage Controller 1	Dell™ Boot Optimized Server Storage (BOSS)-N1	Dell™ Boot Optimized Server Storage (BOSS)-N1	RAID bus controller: Broadcom®/LSI MegaRAID® 12 G SAS/PCIe® Secure SAS39xx, SAS 3908	Broadcom® SAS 3916

Disk	480 GB Dell™ EC NVM Express® (NVMe®) Instant Scramble Erase (ISE) 7400 read-intensive (RI) M.2	480 GB Dell™ NVMe® PE8010 read-intensive M.2	960 GB Samsung® MZQL2960HCJR-00A07	960 GB KIOXIA® KCD6XLUL960G
Number of Disks	2	2	6	6
Storage Controller 2	PCIe® solid-state drive (SSD) backplane	Dell™ PowerEdge RAID Controller 11 (PERC 11) H755 Front	Broadcom® SAS 3908 AOC-S3908L-H8iR	Micron® NVMe® SSD controller
Disk	3.84 TB Dell™ NVMe® CM6 read-intensive	1.75 TB Samsung® PM897 Dell™ MZ7L31T9HBNAAD3 SATA	480 GB Micron® 7450 MTFDKBA480TFR	480 GB Micron® 7450 MTFDKBA480
Number of Disks	4	4	2	2
Storage Controller 3	Not applicable (N/A)	Broadcom® MegaRAID® 12GSAS/PCIe® secure SAS29xx	N/A	N/A
Disk	N/A	3.84 TB Dell™ NVMe® P5500 read-intensive U.2	N/A	N/A
Number of Disks	N/A	4	N/A	N/A
Installed Memory	512 MB	512 MB	512 MB	512 MB
Memory Speed	4,800 megatransfers per second (MT/s)	4,800 MT/s	4,800 MT/s	4,800 MT/s
Number of Memory DIMMs	16 x 32 GB error correction code (ECC) DDR5	16 x 32 GB ECC DDR5	16 x 32 GB ECC DDR5	16 x 32 GB ECC DDR5
Network	Broadcom® and subsidiaries NetXtreme® BCM5720 1 x Broadcom® and subsidiaries BCM57508 NetXtreme® E-Series 10 Gb/25 Gb/40 Gb/50 Gb/100 Gb/200 Gb 1 x Broadcom® and subsidiaries BCM57414 NetXtreme® E-Series	1 x Broadcom® and subsidiaries NetXtreme® BCM5720 1 x Mellonox® MT2892 1 x Broadcom® and subsidiaries BCM57414 NetXtreme® E-Series	1 x Microchip® LAN7500 1 x Broadcom® and subsidiaries BCM57508 NetXtreme® E-Series 10 Gb/25 Gb/40 Gb/50 Gb/100 Gb/200 Gb 1 x Intel® Ethernet Network Adapter E810-XXV for SFP	1 x Broadcom® BCM57508 NetXtreme® E-Series 10 Gb/25 Gb/40 Gb/50 Gb/100 Gb/200 Gb Ethernet 1 x Realtek® RTL8153 Gigabit Ethernet Adapter 1 x Intel® Ethernet Controller E810-XXV for SFP
Operating System (OS)	Red Hat® Enterprise Linux®	Red Hat® Enterprise Linux®	Red Hat® Enterprise Linux®	Red Hat® Enterprise Linux®
OS Version	9.2	9.2	9.2	9.2
OS Kernel	Linux® 5.14.0-284.18.1.el9_2.x86_64	Linux® 5.14.0-284.18.1.el9_2.x86_64	Linux® 5.14.0-284.18.1.el9_2.x86_64	Linux® 5.14.0-284.18.1.el9_2.x86_64
BIOS Version	2.1.0	2.1.0	1.1	1.3

Testing Summary

Prowess Consulting engineers documented the rack and stack process for each of the four servers. The steps included:

- Racking servers with provided rails
- Connecting network and power
- Utilizing a crash cart to capture the IP details to connect to the baseboard management controller (BMC)
- Connecting to the BMC of each server to review the configuration
- Installing the OS
- Documenting the ease of use of each system

We tested the following workloads on each of the four servers:

- MySQL® with HammerDB 4.8
- Virtual desktop infrastructure (VDI)-like deployment utilizing CentOS® and stress-ng to simulate a workload on the virtual machine (VM)

For each test and server, we collected system metrics and power metrics utilizing the following tools:

- Nigel's Monitor (nmon)
- Dstat
- Atop

We collected power metrics on the PowerEdge HS5610 and PowerEdge HS5620 servers utilizing racadm and the following PowerShell® script:

```
$i=1
do  {
        $i=$i++
        Start-Sleep -Seconds 5
        $date=get-date | Out-File <file location> -Append
        racadm -r <idrac IP> -u root -p <idrac password> get system.power | Out-File <file location>
-Append
}while($i = 4500)
```

We collected power metrics on the SuperServer SYS-120C-TN10R and SuperServer SYS-620C-TN12R servers utilizing IPMICFG and the following shell script:

```
while true; do date | tee -a ~/stress_power; ./IPMICFG-Linux.x86_64 -dcmi power | tee -a ~/stress_
power;sleep 2;done
```

Testing Procedures

Firmware Update Process

Dell PowerEdge HS5610

1. Navigate to www.dell.com/support/home/en-us?app=products.
2. Search for **HS5610**, and then select the server's name from the drop-down menu.
3. Select **Drivers and Downloads**.
4. Locate the update the labeled BIOS, and then click the **Download** button to the right.
5. Locate the update the labeled BMC, and then click the **Download** button to the right.
6. Log in to Dell™ OpenManage™ Server Administrator.
7. Expand **Operations**, and then select **Firmware**.
8. Scroll to the bottom of the firmware page, and then click **Add File**.
9. Browse to and select the firmware file.
10. Click **Upload**.
11. When prompted, click **Refresh** to verify the upload.
12. From the **BMC** section, under **Uploaded Image**, click **Activate**.

Dell PowerEdge HS5610

1. Navigate to www.dell.com/support/home/en-us?app=products.
2. Search for **HS5620**, and then select the server's name from the drop-down menu.
3. Select **Drivers and Downloads**.
4. Locate the update the labeled BIOS, and then click the **Download** button to the right.
5. Locate the update the labeled BMC, and then click the **Download** button to the right.
6. Log in to the BMC interface.
7. Select **Operations > Firmware**.
8. Scroll to the bottom of the page.
9. Click **Add File**, and then select the BIOS or BMC file.
10. Click **Upload File**.
11. Wait while it applies.
12. Click **Select to apply now**.
13. Refresh when prompted.
14. Click **Activate** on the **Uploaded Image** section to make it live.

Supermicro® SuperServer® SYS-120C-TN10R and SuperServer SYS-620C-TN12R

1. Log in to the Supermicro BMC.
2. Click **Firmware Updates**.
3. Select **BMC**.
4. Click **Next**.
5. For **Step 2: Select file**, click **Select File**.
6. Browse to and select the firmware update file, and then click **Open**.
7. Click **Upload**.
8. Scroll down to **Step 3: File Version** at the bottom of the page, and then click **Update**.
9. The BMC will reboot when completed.
10. Log in to Supermicro BMC.
11. Click **Firmware Updates**.
12. Select **BIOS**.
13. Under **Choose Update Time**, select **Immediate Update**.
14. Click **Next**.
15. Under **Step 2: Select File**, click **Select File**.
16. Browse to and select the firmware update file, and then click **Open**.
17. Click **Upload**.
18. Scroll down to **Step 3: File Version** at the bottom of the page, and then click **Update**.
19. When prompted that the BIOS firmware update has completed, click **OK**.

MySQL Testing

The following steps outline the process to install Red Hat® Enterprise Linux® and MySQL® Community Server.

PowerEdge Deployment Using OpenBMC

1. PowerEdge servers:
   a. Launch a web browser and browse to the iDRAC IP of the server.
   b. Select **Virtual Console** to launch the virtual console session.
   c. From the virtual console, click **Virtual Media**.
   d. On the **Virtual Media** page, click **Connect Virtual Media**.
   e. From the **Map CD/DVD** section, click **Choose File**.
   f. Search for and select the Red Hat Enterprise Linux ISO, and then click **Open**.
   g. Click **Map Device**, and then click **Close**.
   h. From the virtual console, click **Boot**, select **Virtual CD/DVD/ISO**, and then, when prompted to confirm the boot action, click **Yes**.
   i. From the virtual console, click **Power**, select **reset system**, and then, at the **Confirm Power Action** prompt, click **Yes**.

2. SuperServer servers:
    a. Select **Remote Console** to launch.
    b. From the iKVM virtual console, click **Virtual Media**.
    c. Select **Virtual Storage**.
    d. At the **Device 1** page, from the logical drive type drop-down menu, select **ISO Image**, click **Open Image**, browse to and select the Red Hat installation media, and then click **Open**.
    e. From the virtual media page, click **Plug In**, and then click **OK**.
    f. Click **Power Control**, and then select **Power Reset**.
    g. Press **F1** to enter the boot menu.
3. Select **Install Red Hat Enterprise Linux 9.1**.
4. At the **Welcome to Red Hat Enterprise Linux 9.1** page, click **Continue**.
5. From the **Installation Summary** page, click **Root Password**.
6. At the **Root password** page, enter a **Root password**, confirm the password, and then click **Done**.
7. From the **Installation Summary** page, click **Connect to Red Hat**, enter account information, click **Register**, and then click **Done**.
8. From the **Installation Summary** page, click **Software Selection**.
9. Select **Server**, and then click **Done**.
10. From the **Installation Summary** page, click **Installation Destination**.
    a. From the **Local Standard Disks** section, select **BOSS**.
    b. From the **Storage Configuration** section, select **Custom**.
    c. Click **Done**.
    d. For the **New Red Hat Enterprise Linux 9.1** installation, click **Click here to create them automatically** to automatically create the mount points.
    e. Update the mount points with the following parameters.
        i. **/home: 100**
        ii. **swap: 16**
        iii. **/<root>: remainder of storage**
    f. Click **Update Settings**.
    g. Click **Done**.
    h. When the summary of changes pops up, click **Accept changes**.
11. 11. From the **Installation Summary** page, click **Network & Host Name**.
    a. Ensure all connected Ethernet devices are enabled.
    b. In the **Host Name** field, enter a name.
    c. Click **Apply**.
    d. Click **Done**.
12. Click **Begin Installation**.
13. Click **Reboot System**.
14. Connect to the server using a Secure Shell (SSH) client.
15. Run the following command to install updates:

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

16. Run the following command to create software RAID 5 utilizing the disks attached:

```
mdadm --create /dev/md0 --level=5 --raid-devices=4 /dev/nvme1n1 /dev/nvme2n1 /dev/nvme3n1 /dev/
nvme4n1
```

1. Follow these instructions to install: MySQL :: MySQL 8.0 Reference Manual :: 2.5.1 Installing MySQL on Linux Using the MySQL Yum Repository.
2. Copy the following to **/etc/my.cnf**:

```
# 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,
# this will increase compatibility with older clients. For background, see:
# https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_default_authentication_
  plugin
# default-authentication-plugin=mysql_native_password

#skip-grant-tables
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
# general
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
haracter_set_server=latin1
collation_server=latin1_swedish_ci
transaction_isolation=REPEATABLE-READ
# files
innodb_file_per_table
innodb_log_file_size=1024M
innodb_log_files_in_group=8 #scale
innodb_open_files=4000
# buffers
innodb_buffer_pool_size=24000M #scale
innodb_buffer_pool_instances=16
innodb_log_buffer_size=64M
# tune
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
# perf special
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
# monitoring
innodb_monitor_enable=’%’
```

MySQL Client

1. Install Red Hat Enterprise Linux as prescribed above.
2. Download HammerDB 4.8 for Red Hat Enterprise Linux.
3. Extract HammerDB.
4. Run the following command to launch the HammerDB command-line interface (CLI):

```
./hammerdbcli
```

5. Run the following commands to build the benchmark:

```
dbset db mysql
dbset bm TPROC-C
diset connection mysql_host <ip>
diset tpcc mysql_count_ware 500
diset tpcc mysql_num_vu 50
diset tpcc mysql_user root
diset tpcc mysql_user <password>
buildschema
```

6. Run the following command to back up the database after build completion.

```
mysqldump -u root -p tpcc > /mysql/data/backup/tpcc_backup.sql
```

7. Run the following script to test the workloads:

```
dbset db mysql
dbset db TPROC-C
diset tpcc mysql_total_iterations = 1000000
diset tpcc mysql_driver timed
diset tpcc mysql_rampup 5
diset tpcc mysql_duration 20
diset tpcc mysql_allwarehouse true
diset tpcc mysql_timeprofile true
vuset showoutput 1
vuset logtotemp 1
vuset timestamps 1
vuset unique 1
loadscript
puts “SEQUENCE STARTED”
foreach z { 5 10 20 50 100} {
        puts “$z VU TEST”
        vuset vu $z
        vucreate
        vurun
        vudestroy
        after 1900
}
```

VDI-Like Testing

This is the process we used to conduct the VM capacity testing:

1. From a local console, connect to the Red Hat Enterprise Linux instance via SSH by running the following command:

```
ssh root@$instance_ip
```

2. To prepare a file system on the software raid device, run the following command:

```
mkfs.xfs /dev/md0
```

3. To create the mount point for the volume, run the following command:

```
mkdir /var/lib/libvirt
```

4. To set the volume to auto mount in future runs:
   a. To open the fstab file, run the following command:

```
vim /etc/fstab
```

   b. Press **a**.
   c. To the end of the file, add the following:

```
/dev/md0                /var/lib/libvirt                xfs     defaults        0 1
```

   d. Enter **:wq** to save the changes and exit vim.
5. To mount the volume, run the following command:

```
mount -a
```

6. To install the cockpit components, run the following command:

```
dnf install -y cockpit && systemctl enable --now cockpit.socket && dnf install -y cockpit-machines
virt-viewer qemu-kvm libvirt virt-install
```

7. To set up virt-host info, run the following command:

```
for drv in qemu network nodedev nwfilter secret storage interface; do systemctl start virt${drv}d{,-
ro,-admin}.socket; done
```

8. To enable IOMMU on the kernel, run the following command:

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

9. To reboot the system, run the following command:

```
reboot
```

10. Connect back to the host from your local console via SSH.
11. To validate the virt-host installation, run the following command:

```
vert-host-validate
```

12. From the local system, open a web browser to **$instance_ip:9090** to access the cockpit graphical user interface (GUI).
13. When prompted, enter **root** as the username, and then enter the system password.

14. Select **Virtual machines** from the left-hand menu, and then click **Create VM**.
15. In the resulting window, specify the following:
    a. **Name: CentOS-1**
    b. **Installation Type: Download an OS**
    c. **Operating System: CentOS stream 9**
    d. **Storage: Create a new volume**
    e. **Storage Limit: 25 GB**
    f. **Memory: 8 GB**
16. Select **Create and Run**.
17. Select the VM instance name.
18. Select the expand button on the console on the right-hand side.
19. Select **English**, and then click **Next**.
20. Select root user, set a password, select **Allow root ssh**, and then click **Done**.
21. Select an installation destination, confirm the 25 GB volume, and then click **Done**.
22. Click **Begin installation**.
23. Click **Reboot system** once completed.
24. In a terminal window, connect to the Red Hat Enterprise Linux instance terminal.
25. From the Red Hat Enterprise Linux terminal, connect via SSH to the IP of the CentOS-1 VM.
26. From the CentOS-1 VM terminal, run the following command to update the system:

```
dnf update -y
```

27. To install stress-ng, run the following command:

```
dnf install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm; dnf install -y
stress-ng
```

28. To populate the startup script, run the following command:

```
echo -e “#! /bin/bash \n stress-ng --cpu 2 --io 2 --vm 4 --vm-bytes 1G --timeout 5” > /startup.sh;
chmod +x /startup.sh
```

29. To run the job stress-ng workload via cron:
    a. To edit the crontab, run the following command:

```
crontab -e
```

    b. Press **a**.
    c. Add ***/5 * * * * /startup.sh** to the file.
    d. To save the changes and exit, enter **:wq**.
30. To create a startup service to trigger the stress-ng workload, run the following command:

```
tee>/etc/systemd/system/stress.service <<EOF
[Unit]
 Description=stress test
 After=network.target
[Service]
 Type=simple
 ExecStart=/startup.sh
 TimeoutStartSec=0
[Install]
 WantedBy=default.target
EOF
chmod +xw /etc/systemd/system/stress.service
systemctl enable stress.service
systemctl start stress.service
```

31. To stop the CentOS-1 VM, run the following command:

```
shutdown -h
```

32. To clone the CentOS-1 VM, from the Red Hat Enterprise Linux command line, run the following command:

```
for I in {2..200};do virt-clone -o CentOS-1—-auto-clone -n CentOS-$i; sleep 10; done
```

33. To populate the stress script, run the following command:

```
tee>/stress.sh <<EOF
#! /bin/bash
echo “Run number: “
read runNum
mkdir -p ~/stress_run/\$runNum
nmon -F ~/stress_run/\$runNum/Run_\${runNum}_nmon.csv -s 15 -c 3600
dstat -trdcD md0 —device-mapper -L rhel -l >> ~/stress_run/\$runNum/Run_\${runNum}_dstat.out &
echo “Adding initial 4 VMs”
virsh start CentOS-1
virsh start CentOS-2
virsh start CentOS-3
virsh start CentOS-4
echo “initial VMs added”
sleep 300
for I in {5..100}
do
        echo““Adding VM number: \$””
        date
        virsh start CentOS-\$i
        sleep 300
done
EOF
```

34. To add execution to the script, run the following command:

```
chmod +x /stress.sh
```

35. To open a terminal multiplexer session for the test run, run the following command:

```
tmux
```

36. To split the resulting window into a few screens:
    a. To split the screen in half, press **Ctrl+B**, and then press **“**.
    b. To switch between windows, press **Ctrl+B**, and then press an arrow key.
37. In the largest window, to start a top session, run the following command:

```
top
```

38. To start the testing, run the following command:

```
/stress.sh
```

39. Monitor the progress on the screen until new VMs are no longer being added every 5 minutes, and then make a note of the last VM ID added.

iPerf Network Testing

This section contains the steps to set up the various hosts and then to run the tests against each host.

iPerf Setup

This is the process we used to conduct the network speed testing with iPerf:

1. Download the Ubuntu 22.04 ISO from https://releases.ubuntu.com/22.04/ubuntu-22.04.3-desktop-amd64.iso.torrent?_ga=2.17682619.343639536.1692031889-1815853887.1674678963.
2. Connect to the BMC web console view for the host, and then boot to the Ubuntu ISO.
3. Select **Try or install Ubuntu**.
4. In the resulting GUI, click I**nstall Ubuntu**.
5. Select **Minimal installation**.
6. Select **Install 3rd party software**.
7. Click **Continue**.
8. Select **Erase disk and install Ubuntu**.

9. Click **Continue**.
10. Click **Confirm** to acknowledge the changes to disk.
11. Select your time zone, and then click **Continue**.
12. At the resulting window, set values as appropriate:
    a. **Name:** ***Your name***
    b. **Hostname:** ***System identifying hostname***
    c. **Username:** ***Username***
    d. **Password:** ***Chosen password***
    e. **Confirm password:** ***Chosen password again***
13. Click **Continue**.
14. After the installation completes, click **Restart now**.
15. Log back in with the previously specified username and password.
16. Click the network icon in the top right corner, and then select **Settings**.
17. Locate the **25000 Mbs** connection, and then note the device ID, henceforth **$L_25**.
18. Select the gear icon next to the **25000 Mbs** connection.
19. In the **Identity** tab, set:
    a. **Name: 25gbe**
20. In the **IPv4** tab, set:
    a. **IPv4 Method: Manual**
    b. **Address: As appropriate**
    c. **Netmask: 24**
21. Click **Apply**.
22. Locate the **100000 Mbs** connection, and then note the device ID, henceforth **$L_100**.
23. Select the gear icon next to the **100000 Mbs** connection.
24. In the **Identity** tab, set:
    a. **Name: 100gbe**
25. In the **IPv4** tab, set:
    a. **IPv4 Method: Manual**
    b. **Address: As appropriate**
    c. **Netmask: 24**
26. Click **Apply**.
27. Select the gear icon next to the **1000 Mbs** connection.
28. Make note of the IP address, henceforth **$access_ip**.
29. Click **Cancel**.
30. Click the dial pad in the lower left-hand corner, and then type **Terminal**.
31. Click on the terminal app.
32. In the terminal app, run the following command:

```
sudo apt update; sudo apt install -y openssh-server
```

33. From the local system, to connect to the instance, run the following command:

```
ssh $username@$access_ip
```

34. To switch to root-level access, run the following command:

```
sudo su –
```

35. To install necessary tools on the system, run the following command:

```
apt-get install -y tmux build-essential vim numactl nmon
wget https://downloads.es.net/pub/iperf/iperf-3.13-mt1.tar.gz
tar -xvzf iperf-3.13-mt1.tar.gz
cd iperf-3.13-mt1/
./configure
Make
make install
```

36. To create the script used to launch the server process, run the following command:

```
vim ~/iperf-server.sh
```

   a. Type :**set paste**.
   b. Press **a**.
   c. Paste in the following content:

```
        #! /bin/bash
        # Usage iperf-server.sh 25|100
        L_25=””
        L_100=””

        speed=$1
        iperf=”/root/iperf-3.13-mt1/src/iperf3”
        L_25_name=”25gbe”
        L_100_name=”100gbe”
        systemctl stop irqbalance.service
        nmcli con down $L_25_name
        nmcli con down $L_100_name
        L_25_node=`cat /sys/class/net/${L_25}/device/numa_node`
        L_100_node=`cat /sys/class/net/${L_100}/device/numa_node`
        if [ $speed == “25” ] ; then
                nmcli con mod $L_25_name
                nmcli con up $L_25_name
                sleep 2
                L_25_ip=`nmcli dev show $L_25 | grep IP4.ADD | awk ‘{print $2}’ | awk -F/
‘{print $1}’`
                echo “IP: $L_25_ip”
                numactl -N $L_25_node -l $iperf --server -B $L_25_ip
        elif [ $speed == “100” ]; then
                nmcli con up $L_100_name
                sleep 2
                L_100_ip=`nmcli dev show $L_100 | grep IP4.ADD | awk ‘{print $2}’ | awk -F/
‘{print $1}’`
                echo “IP: $L_100_ip”
                numactl -N $L_100_node -l $iperf --server -B $L_100_ip
        else
                echo “No speed selected, specify either 25 or 100”
fi
```

   d. Update the **L_25** and **L_100** variables to the values noted earlier.
   e. To save and exit, press **[Esc] :wq [Enter]**.

37. To make the script executable, run the following command:

```
chmod +x ~/iperf-server.sh
```

38. To create the script used to launch the client process, run the following command:

```
vim ~/iperf-client.sh
```

   a. Type :**set paste**.
   b. Press **a**.
   c. Paste in the following content:

```
#! /bin/bash
## Usage iperfP.sh IP_TO_TEST
#Vars
L_25=””
L_100=””
####
```

```
server_ip=$1
iperf=”/root/iperf-3.13-mt1/src/iperf3”
L_25_name=”25gbe”
L_100_name=”100gbe”
#Numa
L_25_node=`cat /sys/class/net/${L_25}/device/numa_node`
L_100_node=`cat /sys/class/net/${L_100}/device/numa_node`
#Clean Slate starting Point
systemctl stop irqbalance.service
nmcli con down $L_25_name
nmcli con down $L_100_name
nmcli con up $L_25_name
nmcli con up $L_100_name
sleep 2
#Discovered Vars
TS=$(date +%s)
L_25_ip=`nmcli dev show $L_25 | grep IP4.ADD | awk ‘{print $2}’ | awk -F/ ‘{print $1}’`
L_100_ip=`nmcli dev show $L_100 | grep IP4.ADD | awk ‘{print $2}’ | awk -F/ ‘{print $1}’`
run_dir=./`hostname`_to_${server_ip}_${TS}
sleep 2
killall nmon
mkdir $run_dir
cp $0 $run_dir/
nmon -F ./$run_dir/load_summary.nmon -s 15 -c 60
nmcli dev show $L_25 > ./$run_dir/25GbE_adapter.txt
nmcli dev show $L_100 > ./$run_dir/100GbE_adapter.txt
#25Gbe Test
nmcli con down $L_100_name
mkdir ./$run_dir/25GbE
sleep 2
for P in 1 2 4 8 16; do
        for R in {1..3};do
                echo “starting run $R against $server_ip with $P threads”
                numactl -N $L_25_node -l $iperf -c $server_ip -B $L_25_ip -p 5201 -P $P -i 10s
--logfile ./$run_dir/25GbE/to_${server_ip}_run_${R}_threads_${P}.log
                sleep 15
done
done
#100 GbE test
nmcli con down $L_25_name
sleep 2
nmcli con up $L_100_name
mkdir ./$run_dir/100GbE
sleep 2
for P in 1 2 4 8 16; do
        for R in {1..3};do
                echo “starting run $R against $server_ip with $P threads”
                numactl -N $L_100_node -l $iperf -c $server_ip -B $L_100_ip -p 5201 -P $P -i
10s --logfile .//$run_dir/100GbE/to_${server_ip}_run_${R}_threads_${P}.log
                 sleep 15
         done
done
```

```
##### Parsing
# tail -n4 ./*/*.log | grep sender | cut -dG -f2 | cut -d “ “ -f3
# tail -n4 ./*/*.log | grep reciver | cut -dG -f2 | cut -d “ “ -f3

echo “1P S Gbs, 1P R Gbs, 2P S Gbs, 2P R Gbs, 4P S Gbs, 4P R Gbs, 8P S Gbs, 8P R Gbs” | tee /
dev/tty | tee >> ./$run_dir/run_summary.log
echo -e “ ##### 25 GbE Results” | tee /dev/tty | tee >> ./$run_dir/run_summary.log

for R in 1 2 3 ;do
               for P in 1 2 4 8 16; do
       declare S_rate_$P=`tail -n4 ./$run_dir/25GbE/to_${server_ip}_run_${R}_threads_${P}.log
| grep sender | cut -dG -f2 | cut -d “ “ -f3`
       declare R_rate_$P=`tail -n4 ./$run_dir/25GbE/to_${server_ip}_run_${R}_threads_${P}.log
| grep receiver | cut -dG -f2 | cut -d “ “ -f3`
done
echo “$S_rate_1 , $R_rate_1 , $S_rate_2 , $R_rate_2 , $S_rate_4 , $R_rate_4 , $S_rate_8 , $R_
rate_8 “ | tee /dev/tty | tee >> ./$run_dir/run_summary.log
done
echo -e “ ##### 100 GbE Results” | tee /dev/tty | tee >> ./$run_dir/run_summary.log
for R in 1 2 3 ; do
               for P in 1 2 4 8 16; do
                      declare S_rate_$P=`tail -n4
./$run_dir/100GbE/to_${server_ip}_run_${R}_         threads_${P}.log | grep sender | cut -dG
-f2 | cut -d “ “ -f3`
                      declare R_rate_$P=`tail -n4 ./$run_dir/100GbE/to_${server_ip}_run_${R}_
threads_${P}.log | grep receiver | cut -dG -f2 | cut -d “ “ -f3`
done

echo “$S_rate_1 , $R_rate_1 , $S_rate_2 , $R_rate_2 , $S_rate_4 , $R_rate_4 , $S_rate_8 , $R_
rate_8 “ | tee /dev/tty | tee >> ./$run_dir/run_summary.log
done
```

   d. Update the **L_25** and **L_100** variables to the values noted earlier.
   e. To save and exit, press **[Esc] :wq [Enter]**.

39. To create the script used to launch the client process, run the following command:

```
chmod +x ~/iperf-client.sh
```

40. Repeat this process for each of the four hosts.

iPerf Network Testing

This section outlines the steps to run iPerf in server mode on one system, henceforth “the server,” and then from a second host, henceforth “the client,” running iPerf in client mode and connecting to the server to test network speeds.

1. Open two terminal windows on the local system.
2. In the first terminal window, to connect to the host acting as the server, run the following command:

```
ssh username@$hosts_access_ip
```

3. To start the server for the 25 gigabit Ethernet (GbE) network testing, run the following command:

```
sudo /root/iperf-server.sh 25
```

4. Make note of the IP shown, henceforth **$testing_25_ip**.
5. In the second terminal window, to connect to the host acting as a client, run the following command:

```
ssh username@$hosts_access_ip
```

6. To start the test against the 25 GbE interface of the host system, run the following command:

```
sudo /root/iperf-client.sh $testing_25_ip
```

7. The test results will display on screen and save into a directory in the current path, with a name based on the system host name, test IP, and time stamp.

8. Open the first terminal window again and press **Ctrl+C** to stop the server running on the 25 GbE interface.
9. To start the server on the 100 GbE interface, run the following command:

```
    sudo /root/iperf-server 100
```

10. Make note of the IP shown, henceforth **$testing_100_ip**.
11. In the second terminal window, to start the test against the 100 GbE, run the following command:

```
    sudo /root/iperf-client.sh $testing_100_ip
```

12. Repeat as needed for each combination of hosts.

The analysis in this document was done by Prowess Consulting and commissioned by Dell Technologies.

Results have been simulated and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.