

## Behind the Report—Testing Addendum: Dell™ PowerEdge™ R750 Servers with NVMe® RAID Increase Microsoft® SQL Server® Database Performance

This document provides the system-configuration details and step-by-step procedures that Prowess Consulting used to collect performance data for running a Microsoft® SQL Server® database on a Dell™ PowerEdge™ R750 server.

For the full analysis, read the report, <https://www.prowesscorp.com/project/dell-r750-vs-r740xd/>.

Testing was concluded in May 2022.

|   | Dell™ PowerEdge™ R740xd Server  | Dell™ PowerEdge™ R750 Server  |
|---|---|---|
| Processor   |   |   |
| Processor   | 2 x Intel® Xeon® Gold 6238 processor (2nd Generation Intel® Xeon® Scalable processor) | 2 x Intel® Xeon® Gold 6338 processor (3rd Generation Intel® Xeon® Scalable processor) |
| Processor base frequency  | 2.1 GHz   | 2.0 GHz   |
| Cores/threads   | 22/44   | 32/64   |
| Total cores   | 44  | 64  |
| Memory  |   |   |
| Total memory in system  | 256 GB  | 256 GB  |
| Number and type of memory modules   | 4 x 32 GB Hynix® dual-rank DDR4<br>4 x 32 GB Samsung® dual-rank DDR4                  | 16 x 16 GB Hynix® DDR4 DIMMs  |
| Memory speed  | 2,933 megatransfers per second (MT/s)   | 3,200 megatransfers per second (MT/s)   |
| Controller details for the Microsoft® SQL Server® 2019 database data and log volume |   |   |
| Controller  | 1 x Dell™ PowerEdge RAID Controller 10 (PERC 10) H740P                                | 2 x Dell™ PowerEdge RAID Controller 11 (PERC 11) H755N Front                          |
| Disks   | 16 x 960 GB Samsung® MZ7LH960HBJR0D3 Serial ATA (SATA®) solid-state drives (SSDs)     | 16 x 960 GB Dell™ data-center NVMe Express® (NVMe®) PE8010 RI U.2 (SK hynix)          |
| Controller details for the operating system (OS)                                    |   |   |
| Controller  | Marvell® Technology Group Dell™ Boot Optimized Server Storage (BOSS)-S1               | Dell™ Boot Optimized Server Storage (BOSS)-S2   |
| Disks   | 2 x 240 GB Intel® SSDSCKKB240G8R SATA SSD   | 2 x 480 GB Micron® 5300 PRO SSD   |

| Networking                       |  |  |
|----------------------------------|--|--|
| Network adapters                 | 1 x Broadcom® BCM57416 NetXtreme® E-series dual-media 10 GbE RDMA Ethernet controller<br><br>1 x Broadcom® NetXtreme® BCM5720 GbE PCIe<br><br>1 x 25 GbE dual-port Intel® Ethernet Converged Network Adapter E810-XXV  | 1 x 1 gigabit Ethernet (GbE) dual-port Broadcom® BCM5720 NetXtreme® Ethernet controller<br><br>1 x 10/25 GbE dual-port Broadcom® BCM57414 NetXtreme® E-series remote direct memory access (RDMA) Ethernet controller |
| Server details                   |  |  |
| BIOS version                     | BIOS 2.13.3  | BIOS 1.5.5   |
| OS name and version <sup>1</sup> | Red Hat® Enterprise Linux® 8.5   | Red Hat® Enterprise Linux® 8.6   |
| OS kernel                        | 4.18.0-348.23.1.el8_5.x86_64   | 4.18.0-372.9.1.el8.x86_64  |
| Database software                |  |  |
| Microsoft® SQL Server®           | Microsoft® SQL Server® 2019 (RTM-CU15) (KB5008996)—15.0.4198.2 (X64)<br><a href="http://www.intel.com/content/www/us/en/products/sku/193949/intel-xeon-gold-6238-processor-30-25m-cache-2-10-ghz/specifications.html">www.intel.com/content/www/us/en/products/sku/193949/intel-xeon-gold-6238-processor-30-25m-cache-2-10-ghz/specifications.html</a> |  |

**Note:** The Dell PowerEdge R750 server testing was performed on bare-metal servers with no hypervisor/virtualization layer.

#### Testing Procedures

The following sections describe the step-by-step procedures used to configure the systems, verify the configurations, and run the benchmark tests.

#### RAID Configuration

Deploy a Dell PowerEdge R750 server with sixteen 16 x 960 GB Dell™ data-center NVM Express® (NVMe®) PE8010 RI U.2 (SK hynix) SSDs connected equally to two Dell™ PowerEdge RAID Controller 11 (PERC 11) H755N Front RAID controllers with the following configuration:

SQL\_Data virtual disk:

- RAID configuration: RAID 5
- Number of disks: 8
- Virtual disk properties:
  - Strip size = 256 KB
  - Number of blocks = 43750260736
  - Span depth = 1
  - Number of drives per span = 8
  - Write cache (initial setting) = WriteThrough
  - Disk cache policy = Disabled
  - Encryption = None
  - Data protection = None
  - Active operations = None
  - Exposed to OS = Yes
  - Emulation type = Default
  - Cachebypass size = Cachebypass-64k
  - Cachebypass mode = Cachebypass Intelligent

SQL\_Log virtual disk:

- RAID configuration: RAID 10
- Number of disks: 8
- Virtual disk properties:
  - Strip size = 256 KB
  - Number of blocks = 25000148992
  - Span depth = Not applicable (N/A)
  - Number of drives per span = N/A
  - Write cache (initial setting) = WriteThrough
  - Disk cache policy = Disabled
  - Encryption = None
  - Data protection = None
  - Active operations = None
  - Exposed to OS = Yes
  - Emulation type = Default
  - Cachebypass size = Cachebypass-64k
  - Cachebypass mode = Cachebypass Intelligent
  - Is LD ready for OS requests = Yes

Deploy a Dell PowerEdge R740xd server with sixteen 960 GB Dell Serial ATA (SATA®) PM883a read-intensive SSDs connected to one Dell PERC 10 H740P RAID Controller with the following configuration:

SQL\_Data virtual disk:

- RAID configuration: RAID 5
- Number of disks: 8
- Virtual disk properties:
  - Strip size = 256 KB
  - Number of blocks = 43750260736
  - Span depth = 1
  - Number of drives per span = 8
  - Write cache (initial setting) = WriteThrough
  - Disk cache policy = Disabled
  - Encryption = None
  - Data protection = None
  - Active operations = None
  - Exposed to OS = Yes
  - Emulation type = Default
  - Cachebypass size = Cachebypass-64k
  - Cachebypass mode = Cachebypass Intelligent

SQL\_Log virtual disk:

- RAID configuration: RAID 10
- Number of disks: 8
- Virtual disk properties:
  - Strip size = 256 KB
  - Number of blocks = 25000148992
  - Span depth = N/A
  - Number of drives per span = N/A
  - Write cache (initial setting) = WriteThrough
  - Disk cache policy = Disabled
  - Encryption = None
  - Data protection = None
  - Active operations = None
  - Exposed to OS = Yes
  - Emulation type = Default
  - Cachebypass size = Cachebypass-64k
  - Cachebypass mode = Cachebypass Intelligent
  - Is LD ready for OS requests = Yes

## Red Hat® Enterprise Linux® Installation

Install Red Hat® Enterprise Linux® 8.5 and SQL Server 2019:

1. Download the latest Red Hat Enterprise Linux 8.5 installation media.
 

**Note:** If you do not already have a Red Hat Enterprise Linux subscription, you can get one by creating a [developer](#) account with Red Hat.

  - Create a boot USB drive from which to install.
2. Install Red Hat Enterprise Linux:
  - a. Select **Time & Date**, and then set the time zone to **Central**.
  - b. Set the **Root Password**.
  - c. Select **Installation Source**.
  - d. Select **Server**.
  - e. Select **Installation Destination**.
  - f. Select the **Dell BOSS** drives as the installation location.
  - g. Select **Subscription**, and then fill in the details.
  - h. Select **Install**.
3. Download and install the [Linux PERCCLI](#) utility.
4. Enter the following commands to ensure that the correct device drive is being configured for the correct volume name:

```
./percccli64 show all
./percccli64 /c0 show
```

5. Locate the virtual disk (VD) list.
6. Note the **VD ID** for each drive name.
7. On the Dell PowerEdge R750 server, set the write cache using the following commands:

```
./percccli64 /c0/v239 set wrcache=wt
./percccli64 /c1/v239 set wrcache=wt
```

8. For each VD, enter:

```
./percccli64 /c0 /v<VD ID> show all
```

9. Copy the **SCSI NAA ID** value.
10. Enter the following command to get the corresponding worldwide name (WWN) values for each drive:

```
ls -la /dev/disk/by-id/ | grep <SCSI NAA ID>
```

11. Locate the device ID.

12. Format the drives with the following file systems:

- SQL\_Data:

```
mkfs.xfs /dev/<device id> -f -L datavolume
```

- SQL\_Log:

```
mkfs.xfs /dev/<device id> -f -L logvolume
```

13. Run the following commands to optimize the disks:

- Create a script and add the following commands:

```
echo none > /sys/block/$1/queue/scheduler
echo 0 > /sys/block/$1/queue/add_random
echo 2 > /sys/block/$1/queue/nomerges
echo 2048 > /sys/block/$1/queue/nr_requests
echo 0 > /sys/block/$1/queue/rotational
echo 2 > /sys/block/$1/queue/rq_affinity
echo 1024 > /sys/block/$1/queue/max_sectors_kb
echo 1024 > /sys/block/$1/device/queue_depth
echo 1 > /sys/block/$1/queue/iostats
```

- Save the script and upload to the server.
- Run the following script:

```
./script.sh <device id>
```

## SQL Server 2019 Installation and Configuration

1. Run the following commands to install prerequisites:

```
sudo yum install python2 compat-openssl10
sudo alternatives --config python
```

2. Run the following commands to enable Tuned and install the SQL Server Tuned profile:

```
systemctl enable tuned
dnf install tuned-profiles-mssql
```

3. The tuned.conf file in /usr/lib/tuned/mssql has the following configuration:

```
#
# tuned configuration
#

[main]
summary=Optimize for MS SQL Server
include=throughput-performance

[vm]
transparent_hugepage.defrag=always

[sysctl]

vm.max_map_count=800000
kernel.numa_balancing=0
kernel.sched_latency_ns=60000000
kernel.sched_min_granularity_ns=15000000
kernel.sched_wakeup_granularity_ns=2000000
```

4. Run the following command to set the SQL Server repository:

```
sudo curl -o /etc/yum.repos.d/mssql
server.repo https://packages.microsoft.com/config/rhel/8/mssql-server-2019.repo
```

5. Run the following command to install SQL Server:

```
sudo dnf install -y mssql-server
```

6. Run the following command to configure SQL Server:

```
/opt/mssql/bin/mssql-conf setup
```

- a. Select **1** for evaluation.
- b. Enter **Yes** to accept the license terms.
- c. Enter a **SQL Server Admin Password**.

7. Run the following command to verify that SQL Server is running:

```
systemctl status mssql-server_
```

8. For testing purposes only, stop and disable the firewall service with the following commands:

```
systemctl stop firewalld
systemctl disable firewalld
```

9. For testing purposes only, disable SELINUX with the following comments:

```
vi /etc/selinux/config

SELINUX=disabled
```

10. Install the SQL Server tools with the following commands:

```
curl -o /etc/yum.repos.d/msprod.repo https://packages.microsoft.com/config/
rhel/8/prod.repo

dnf install -y mssql-tools unixODBC-devel
```

11. Configure `.bash_profile` and `.bashrc` to source the tools and SQL Server install paths using the following commands:

```
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile

echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc

echo 'export PATH="$PATH:/opt/mssql/bin"' >> ~/.bash_profile

echo 'export PATH="$PATH:/opt/mssql/bin"' >> ~/.bashrc

source ~/.bashrc
```

12. Test connectivity to SQL Server by running the following command:

```
sqlcmd -s localhost -U SA -P <sa password>
```

13. Enter the following SQL command to verify the version:

```
Select @@version

go
```

14. Enable trace flag 3979 to support SQL Server and the Forced Unit Access (FUA) input/output (I/O) subsystem using the following commands:

- a. Enable **traceflag 3979**:

```
mssql-conf traceflag 3979 on
```

- b. Set **control.writethrough** in the **mssql-conf configuration** option to **1**:

```
mssql-conf set control.writethrough 1
```

- c. Set **control.alternatewritethrough** in the **mssql-conf configuration** option to **0**:

```
mssql-conf set control.alternatewritethrough 0
```

15. Run the following commands to complete setup of SQL Server:

```
mssql-conf set telemetry.customerfeedback false

sysctl -w kernel.numa_balancing=0

sysctl -w vm.max_map_count=262144

mssql-conf set network.tlsprotocols 1.2
```

16. Set SQL Server memory to 90 percent of available memory with the following command:

```
mssql-conf set memory.memorylimitmb <memory>
```

17. Run the following command to create the SQL Server directory:



```
mkdir -p mssql/data mssql/log/log mssql/log/tempdb
```

- Run the following command to change ownership of the newly created directories:

```
sudo chown mssql:mssql /mssql/data
sudo chown mssql:mssql /mssql/log
```

18. Run the following command to enable execution on the directories:

```
sudo chmod 777 /mssql/data
sudo chmod 777 /mssql/log
```

19. Run the following command mount the file system

```
mount /dev/<device ID> /mssql/data
mount /dev/<device ID> /mssql/log
```

20. Run the following command to get the universally unique identifier (UUID) of each mounted disk:

```
blkid
```

Note the UUID for **datavolume** and **logvolume**.

21. Update **/etc/fstab** with the following command:

```
UUID=<UUID datavolume> /mssql/data xfs rw,attr2,noatime 0 0
UUID=<UUID logvolume> /mssql/log xfs rw,attr2,noatime 0 0
```

22. Update the SQL Server configuration data and log file locations with the following commands:

```
mssql-conf set filelocation.defaultdatadir /mssql/data/
mssql-conf set filelocation.defaultlogdir /mssql/log/log
```

23. Restart the SQL Server service with the following command:

```
systemctl restart mssql-server.service
```

24. Launch the SQL Server management console from a client system.

25. Connect to the Linux SQL Server instance.

26. Run the following commands to modify the location of TempDB:

```
ALTER DATABASE tempdb MODIFY FILE
    (NAME = tempdev, FILENAME = '/mssql/log/tempdb/tempdb01.mdf', SIZE =
    1024, FILEGROWTH = 8192MB)

GO

ALTER DATABASE tempdb MODIFY FILE
    (NAME = templog, FILENAME = '/mssql/tempdb/templog.ldf', SIZE = 1024,
    FILEGROWTH = 8192MB)
```

```
GO
ALTER DATABASE tempdb REMOVE FILE tempdev2
GO
ALTER DATABASE tempdb REMOVE FILE tempdev3
GO
ALTER DATABASE tempdb REMOVE FILE tempdev4
GO
ALTER DATABASE tempdb REMOVE FILE tempdev5
GO
ALTER DATABASE tempdb REMOVE FILE tempdev6
GO
ALTER DATABASE tempdb REMOVE FILE tempdev7
GO
ALTER DATABASE tempdb REMOVE FILE tempdev8
GO

ALTER DATABASE tempdb

ADD FILE (NAME = tempdev2, FILENAME = '/mssql/tempdb/tempdb02.ndf', SIZE = 1024,
FILEGROWTH = 8192MB)

ALTER DATABASE tempdb

ADD FILE (NAME = tempdev3, FILENAME = '/mssql/tempdb/tempdb03.ndf', SIZE = 1024,
FILEGROWTH = 8192MB)

ALTER DATABASE tempdb

ADD FILE (NAME = tempdev4, FILENAME = '/mssql/tempdb/tempdb04.ndf', SIZE = 1024,
FILEGROWTH = 8192MB)

ALTER DATABASE tempdb

ADD FILE (NAME = tempdev5, FILENAME = '/mssql/tempdb/tempdb05.ndf', SIZE = 1024,
FILEGROWTH = 8192MB)

ALTER DATABASE tempdb

ADD FILE (NAME = tempdev6, FILENAME = '/mssql/tempdb/tempdb06.ndf', SIZE = 1024,
FILEGROWTH = 8192MB)

ALTER DATABASE tempdb

ADD FILE (NAME = tempdev7, FILENAME = '/mssql/tempdb/tempdb07.ndf', SIZE = 1024,
FILEGROWTH = 8192MB)

ALTER DATABASE tempdb

ADD FILE (NAME = tempdev8, FILENAME = '/mssql/tempdb/tempdb08.ndf', SIZE = 1024,
FILEGROWTH = 8192MB)
```

27. Download and install Atop to capture system performance information using the following commands:

```
wget https://www.atoptool.nl/download/atop-2.6.0-1.el8.x86_64.rpm
chmod +x atop-2.6.0-1.el8.x86_64.rpm
rpm -ivh atop-2.6.0-1.el8.x86_64.rpm
service atop start
```

28. Capture Atop data using the following command:

```
atop -r -b <beginning time> -e <ending time> > /tmp/pass#.txt
```

29. Capture DSTAT data using the following command:

```
dstat -trdld total,sda,sdb,sdc 60 --output /tmp/pass#.csv
```

## Benchmark Testing New Orders per Minute (NOPM)<sup>1</sup>

After software is installed and pretesting has verified proper configurations, run the benchmark tests to compare the performance between the systems. Run the following workload three times, taking the median of the scores.

### Configure the Windows® 10/Windows Server® 2019 Client Test Machine

1. Configure an Open Database Connectivity (ODBC) data source for the Linux SQL Server:
  - a. Launch Microsoft® ODBC Data Source Administrator (32-bit).
  - b. Click **Add**.
  - c. Select **SQL Server Native Client 11.0**, and then click **Finish**.
  - d. In the **Create a New Data Source to SQL Server** window, in the **Name** field, enter a unique name to identify the Linux SQL Server.
  - e. In the **Server** field, enter a name or IP address to connect to the Linux SQL Server, and then click **Next**.
  - f. In the **How should SQL Server verify the authenticity of the login ID** field, select **With SQL Server authentication**.
  - g. In the **Login ID** field, enter **SA**.
  - h. In the **Password** field, enter the SA password.
  - i. Click **Next**.
  - j. Click **Next** again.
  - k. Click **Finish**.
  - l. Click **Test Data Source** to verify connectivity.
2. Download and install the following SQL Server client tools:
  - a. Client Tools Connectivity
  - b. Client Tools Backwards Compatibility
  - c. Client Tools SDK

3. Copy the provided **MSTPCC.470.tar** file to the testing client.
4. Unzip the .tar file to **C:\MSTPCC.470**.
5. In File Explorer, browse to **C:\MSTPCC.470\SETUP\scripts\1400.war\database**.
6. Open **createdb.sql**, and then update the **FILENAME** path for each file.
7. In File Explorer, browse to **C:\MSTPCC.470\BC-TPCC-v1.6.0**.
8. Double-click **BenchCraft-TPCC.msi** to install BenchCraft®, accepting the default settings.<sup>2</sup>
9. Open a command prompt or PowerShell® session.
10. Run the following command to build the test database:

```
.\setup.cmd /s:<IP or Hostname> /U:sa /P:<password> /W:1400 /B:full /D:normal  
/v:true
```

11. Once the database build is complete, take a backup of the database.
12. Launch the BenchCraft application:
  - a. Click **New Profile**.
  - b. Click **Browse**, and then select the **TPCC\_SQLServer.bdll** plugin file.
  - c. Click **Next**.
  - d. Select **ThreadGroup-1**, and then enter the following values for each setting:
    - **Thread Count: 100**
    - **W\_ID Range End: 1400**
    - **W\_ID Maximum: 1400**
    - **Server Name or IP:** IP or name of the Linux SQL Server
    - **User Name: SA**
    - **Password:** *SA password*
13. Save the profile.
14. Select the **Drivers** page.
15. Select **Launch**.
16. Wait five minutes before pressing F1 to run the load.
17. Run the load for 45 minutes or until the load is at a steady state, and then take the last 15 minutes of the load as the scored value.
18. Once the test is complete, delete and restore the database.
19. Double-click the saved profile to launch the BenchCraft workload utility to run the workload again.

<sup>1</sup>Red Hat Linux 8.6 released May 10, 2022, during Prowess testing. (See <https://access.redhat.com/articles/3078>). As a result, Prowess tested the Dell PowerEdge R750 server with Red Hat Linux 8.6 and the Dell PowerEdge R740xd server with Red Hat Linux 8.5. Prowess reviewed the release notes for Red Hat Linux 8.6 and determined that nothing in the release would affect the testing results (See release notes: [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html/8.6\\_release\\_notes/index](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/8.6_release_notes/index))

<sup>2</sup>Note that this workload is derived from the TPC-C® benchmark and is not comparable to published TPC-C benchmark results, as this implementation does not comply with all requirements of the TPC-C benchmark.



The analysis in this document was done by Prowess Consulting and commissioned by Dell. 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. Prowess and the Prowess logo are trademarks of Prowess Consulting, LLC. Copyright © 2022 Prowess Consulting, LLC. All rights reserved. Other trademarks are the property of their respective owners.