PROWESS

Behind the Report: Can Newer Dell EMC™ Servers Offer Significantly Better Performance for Microsoft® SQL Server®?

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 two Dell Technologies™ platforms:

- Dell EMC™ PowerEdge™ R740xd
- Dell EMC PowerEdge R750

For the full analysis, read the report "Can Newer Dell EMC™ Servers Offer Significantly Better Performance for Microsoft SQL Server?"

Testing was concluded in July 2021.

Server Configurations

	Dell EMC™ PowerEdge™ R740xd	Dell EMC PowerEdge R750
Processor		
Processor	2 x Intel® Xeon® Gold 6230 processor	2 x Intel Xeon Gold 6330 processor
Processor base frequency	2.1 GHz	2.0 GHz
Cores/threads	20/40	28/56
Total cores	40	56
Memory		
Total memory in system	768 GB	1 TB
Number and type of memory modules	24 x 32 GB Micron® DDR4 DIMMs	16 x 64 GB Samsung® M393A8G40AB2-CWE DDR4 DIMMs
Memory speed	2,933 megatransfers per second (MT/s)	3,200 MT/s
RAID array 1 (Microsoft® SQL Server® 2019 database data and log volumes)		
Controller	Dell™ PowerEdge RAID Controller (PERC) H740P	Dell PERC H755N Front
Disks	8 x 1.92 TB Intel® SSD D3-S4510 (Serial ATA [SATA])	8 x 1.6 TB Samsung PM1735 (NVM Express® [NVMe®])
RAID array 2 (boot volume)		
Controller	Dell EMC™ Boot Optimized Server Storage (BOSS)-S1	Dell EMC BOSS-S2

Disks	2 x 240 GB Dell M.2 multi-level cell (MLC) SATA 6 Gbps solid-state drive (SSD)	2 x 480 GB Micron M.2 NVMe SSD
Networking		
Network adapters	10G, dual-media Broadcom® BCM57416 NetXtreme® E-Series Ethernet Controller 25 Gb, dual-port Intel® Ethernet Network Adapter E810-XXV	Broadcom BCM5720 Quad 25 gigabit Ethernet (GbE) Dell EMC PowerEdge Rx5xx LAN on motherboard (OCP3.0) Broadcom adapter
Server details		
BIOS version	2.10.2	1.1.1
Operating system (OS) name and version	Red Hat® Enterprise Linux® 8.4	Red Hat Enterprise Linux 8.4
OS kernel	Linux 4.18.0-305.3.1.el8_4.x86_64	Linux 4.18.0-305.3.1.el8_4.x86_64
Date of last OS updates/patches applied[1]	6/15/2021	6/12/2021
Database software		
Microsoft SQL Server	Microsoft SQL Server 2019 (RTM-CU11) (KB5003249)—15.0.4138.2 (X64)	Microsoft SQL Server 2019 (RTM-CU11) (KB5003249)—15.0.4138.2 (X64)

Note: The Dell EMC PowerEdge R750 and Dell EMC PowerEdge R740xd 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 EMC PowerEdge R750 server with eight 1.6 TB Dell EMC PCIe® Gen4 NVM Express® (NVMe®) SSD drives and the following configuration:

SQL_Data virtual disk:

- RAID configuration: RAID 5
- Number of disks: 3
- Virtual disk properties:
    - Strip size = 256 KB
    - Number of blocks = 6248988672
    - Span depth = 1
    - Number of drives per span = 3
    - Write cache (initial setting) = WriteBack

- 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: 4
- Virtual disk properties:
  - Strip size = 256 KB
  - Number of blocks = 6248988672
  - Span depth = Not applicable (N/A)
  - Number of drives per span = N/A
  - Write cache (initial setting) = WriteBack
  - 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

SQL_TempDB:

- RAID configuration: RAID 0
- Number of disks: 1

- Virtual disk properties:
    - Strip size = 256 KB
    - Number of blocks = 3124494336
    - Span depth = 1
    - Number of drives per span = 1
    - Write cache (initial setting) = WriteBack
    - Disk cache policy = Disk's default
    - 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 EMC PowerEdge R740xd server with eight 1.92 TB Intel® SSD D3-S4510 series 2.5-in SATA drives and the following configuration:

SQL_Data virtual disk:

- RAID configuration: RAID 5
- Number of disks: 3
- Virtual disk properties:
    - Strip size = 256 KB
    - Number of blocks = 7498891264
    - Span depth = 1
    - Number of drives per span = 3
    - Write cache (initial setting) = WriteBack
    - Disk cache policy = Disk's default
    - 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

SQL_Log virtual disk:

- RAID configuration: RAID 10
- Number of disks: 4
- Virtual disk properties:
  - Strip size = 256 KB
  - Number of blocks = 7498891264
  - Span depth = N/A
  - Number of drives per span = N/A
  - Write cache (initial setting) = WriteBack
  - Disk cache policy = Disk's default
  - 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

SQL_TempDB:

- RAID configuration: RAID 0
- Number of disks: 1
- Virtual disk properties:
  - Strip size = 256 KB
  - Number of blocks = 3749445632

- Span depth = 1
- Number of drives per span = 1
- Write cache (initial setting) = WriteBack
- Disk cache policy = Disk's default
- 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® 8.4 and SQL Server 2019:

1. Create a developer account with Red Hat.
2. Download the latest Red Hat Enterprise Linux 8.4 installation media.
3. Create a boot USB drive from which to install.
4. 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 **BOSS** drives as the installation location.
    g. Select **Subscription**, and then fill in the details.
    h. Select **Install**.
5. Download and install the Linux PERCCLI.
6. Enter the following commands to ensure that the correct device drive is being configured for the correct volume name:

```
            ./perccli64 /c0 show
```

7. Locate the virtual disk (VD) list.

8. Note the VD ID for each drive name.

9. On the Dell EMC PowerEdge R750 server, set the write cache using the following commands:

```
./perccli64 /c0/v239 set wrcache=wb
./perccli64 /c0/v238 set wrcache=wb
./perccli64 /c0/v237 set wrcache=wb
```

10. For each VD, enter:

```
./perccli64 /c0 /v<VD ID> show all
```

11. Copy the **SCSI NAA ID value.**

12. 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>
```

13. Locate the device ID.

14. Format the drives with the following file system:

    SQL_Data:

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

    SQL_Log:

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

    SQL_TempDB:

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

15. Run the following command to update the OS:

```
dnf update
```

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 Microsoft 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:

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

9. For testing purposes only, disable SELINUX:

```
vi /etc/selinux/config
SELINUX=disabled
```

10. Install the SQL Server tools:

```
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:

```
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 commands:

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

13. Enter the following SQL command to verify version:

```
Select @@version
go
```

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

    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 the SQL Server memory to 90 percent of available memory:

```
mssql-conf set memory.memorylimitmb 943718
```

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

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

18. Run the following command to get the UUID of each mounted disk:

```
blkid
```

- Note the UUID for **datavolume**, **logvolume**, and **tempdb**.

19. Update /etc/fstab with the following details:

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

20. Update the SQL Server configuration data and log file locations:

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

21. Restart the SQL Server service:

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

22. Launch SQL Server Management Console from a client system.

23. Connect to the Linux SQL Server instance.

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

```
ALTER DATABASE tempdb MODIFY FILE
   (NAME = tempdev, FILENAME = '/mssql/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 '/var/opt/mssql/data/tempdb02.ndf'
GO
ALTER DATABASE tempdb REMOVE FILE '/var/opt/mssql/data/tempdb03.ndf'
GO
ALTER DATABASE tempdb REMOVE FILE '/var/opt/mssql/data/tempdb04.ndf'
GO
ALTER DATABASE tempdb REMOVE FILE '/var/opt/mssql/data/tempdb05.ndf'
GO
ALTER DATABASE tempdb REMOVE FILE '/var/opt/mssql/data/tempdb06.ndf'
GO
ALTER DATABASE tempdb REMOVE FILE '/var/opt/mssql/data/tempdb07.ndf'
GO
ALTER DATABASE tempdb REMOVE FILE '/var/opt/mssql/data/tempdb08.ndf'
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)
```

25. Download and install ATOP to capture system performance information:

```
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
```

26. Capture ATOP data using the following command:

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

27. 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)[2]

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 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.

9. Open a command prompt or PowerShell session.

10. Run the following command to build a TPC-C® 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 TPC-C 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 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 Linux SQL Server
    - **User Name**: SA
    - **Password**: SA password

13. Save the profile.

14. Select the **Drivers** page.

15. Select **Launch**.

16. Wait 5 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 TPC-C database.

19. Double-click the saved profile to launch the Benchcraft workload utility to run the workload again.

[1] Servers were configured and tested serially, which resulted in updates/patches being applied on different days to achieve the same patch version.

[2] Note: Benchmark results are not comparable to standard TPC-C® benchmark tests.

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. Other trademarks are the property of their respective owners.