



The science behind the report:

Handle a higher rate of Microsoft SQL Server transactions with Google Cloud C3 high CPU instances enabled by 4th Gen 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 [Handle a higher rate of Microsoft SQL Server transactions with Google Cloud C3 high CPU instances enabled 4th Gen Intel Xeon Scalable processors](#).

We concluded our hands-on testing on July 21, 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 19, 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: HammerDB results, in new orders per minute, that the N2 and C3 instances achieved. Higher numbers are better. Source: Principled Technologies.

	Storage type	NOPM
n2-highcpu-80	Hyperdisk Extreme	755,012
c3-highcpu-88	Hyperdisk Extreme	1,025,751
C3 vs. N2 % increase	-	35.8%

System configuration information

Table 2: Detailed information on the systems we tested.

General information	
Date testing ended	07/21/2023
Cloud service provider (CSP)	Google Cloud™
Region	us-east-1b

Table 3: Detailed information on the systems we tested.

Workload information	
Workload name and version	HammerDB v4.7 TPROC-C
Workload or software specific parameters	SQL Server: MAXDOP 0, 90% Reserved SQL Memory, MSSQL tuned profile
Iterations and result choice	3 test runs, median reported

Table 4: Detailed information on the systems we tested.

System configuration information	Intel c3-highcpu-88	Intel n2-highcpu-80
Number of instances	1	1
Instance size	c3-highcpu-88	n2-highcpu-80
BIOS name and version	Google Google, 07/12/2023	Google Google, 07/12/2023
vCPU	88	80
Number of cores/threads	44/88	20/40 (36/72)
Memory (GB)	176	80
Underlying processor model	Intel® Xeon® Platinum 8481C @ 2.70GHz	Intel Xeon Platinum 8373C CPU @ 2.60GHz
vNIC information/underlying NIC speed	Google Ethernet Adapter, 100/100 Gbps	Google VirtIO Ethernet Adapter, 100/100 Gbps
Operating system information		
Image or template name and UUID	Ubuntu® 20.04 LTS	Ubuntu 20.04 LTS
Operating system name	Ubuntu 20.04.6 LTS x86/64	Ubuntu 20.04.6 LTS x86/64
Operating system build number	20.04.6	20.04.6
Date patches last applied	07/19/2023	07/19/2023
Changes made from CSP image	Installed Microsoft SQL for Linux® 2022, applied MSSQL tuned profile	Installed Microsoft SQL for Linux 2022, applied MSSQL tuned profile

System configuration information	Intel c3-highcpu-88	Intel n2-highcpu-80
Instance storage (volume type 1)		
Number of volumes	1	1
Volume use in this test	Database data	Database data
CSP volume type	Hyperdisk Extreme	Hyperdisk Extreme
Volume size (GB)	1,000	1,000
IOPS requested	200,000	200,000
Throughput requested	N/A	N/A
Encryption type	Google-managed	Google-managed
Instance storage (volume type 2)		
Number of volumes	1	1
Volume use in this test	Database logs	Database logs
CSP volume type	SSD Persistent	SSD Persistent
Volume size (GB)	60	60
IOPS requested	N/A	N/A
Throughput requested	N/A	N/A
Encryption type	Google-managed	Google-managed

How we tested

Testing overview

For this project, we compared the database performance of Google Cloud instances featuring 4th Generation Intel Xeon Scalable processors vs. 3rd Gen Intel Xeon Scalable processor (Ice Lake) versions. We ran a TPROC-C workload on Microsoft SQL Server hosted on GCP to show that customers choosing updated instances with newer processors can expect an increase in transactions per minute on transactional databases. These steps assume that you already have a database backup in place.

Uploading your database backup to Google Cloud Storage

1. Open a browser, and log into Google Cloud.
2. Click Cloud Storage, and click Buckets.
3. Click Create.
4. Name the bucket, and give it a label.
5. Under Location type, select Region, and choose the region you will be working in.
6. Select a storage class for the data.
7. Click Create.
8. Click into the newly created Bucket, and click Upload Files.
9. Navigate to your database backup folder, and select the backup file.

Creating the Microsoft Windows Server 2022 client image

Creating the baseline client image for instances

1. Log into Google Cloud, and click Go to console.
2. Click Compute engine, and click VM instances.
3. Click Create.
4. In the left window, select New VM instance.
5. Add the following information:
 - Name: Name your instance.
 - Labels: Use any appropriate labels.
 - Region: Select your desired region.
 - Zone: Select your desired zone.
 - Machine family: General-purpose
 - Series: E2
 - Machine type: e2-micro
 - CPU platform: Automatic
 - Boot Disk, and click Change.
 - Operating System: Windows Server
 - Version: Windows Server 2022 Datacenter
 - Boot disk type: SSD persistent disk
 - Size: 50GB
 - Click Select.
 - Identity and API access: App Engine default service account.
 - Firewall: Check Allow HTTP traffic and Allow HTTPs traffic.
6. Click Create.

Configuring Windows Server Datacenter Edition 2022

1. Open Server Manager, and click Local Server.
2. Disable IE Enhanced Security Configuration.
3. Disable Windows Defender Firewall for Domain, Private, and Public networks.
4. Change the time zone to your local time zone.
5. Change the name of your server, and reboot when prompted.
6. Open Server Manager again, and click Local Server.
7. Click to run updates.
8. Run updates, rebooting when prompted, until the server shows no new updates to install.

Installing HammerDB 4.7

1. Download HammerDB from <https://hammerdb.com/download.html>.
2. Double-click the .msi file, choose English, and click OK.
3. Click Yes.
4. Click Next.
5. Chose a destination location, and click Next.
6. Click Next.
7. Click Finish.

Installing Microsoft ODBC Driver 18 for SQL Server

1. Download the latest ODBC Driver for SQL Server from <https://learn.microsoft.com/en-us/sql/connect/odbc/download-odbc-driver-for-sql-server?view=sql-server-ver16>.
2. Double-click the .exe file, and click Next.
3. Check the box to accept the terms, and click Next.
4. Click Next.
5. Click Install.
6. Click Finish.

Creating a machine image of your baseline client instance

1. Log into Google Cloud, and click Go to console.
2. Click Compute engine, and click VM instances.
3. Select the baseline client VM, and click More Actions (the three-dot menu on the right).
4. Click Create new machine image.
5. Give the image a name and optionally a description.
6. Under Location, select Regional, and choose the region you are working in.
7. Leave the Encryption setting default.
8. Click Create.

Creating the Ubuntu 20.04 LTS SUT image

Creating the baseline SUT image for instances

1. Log into Google Cloud, and click Go to console.
2. Click Compute engine, and click VM instances.
3. Click Create.
4. In the left window, select New VM instance.
5. Add the following information:
 - Name: Name your instance.
 - Labels: Use any appropriate labels.
 - Region: Select your desired region.
 - Zone: Select your desired zone.
 - Machine family: General-purpose
 - Series: E2

- Machine type: e2-micro
- CPU platform: Automatic
- Boot Disk, click Change.
 - Operating System: Ubuntu
 - Version: Ubuntu 20.04 LTS
 - Boot disk type: SSD persistent disk
 - Size: 30GB
 - Click Select.
- Identity and API access: App Engine default service account.
- Firewall: Check Allow HTTP traffic and Allow HTTPs traffic.

6. Click Create.

Adding data and volume disks to the baseline instance

1. Log into Google Cloud, and click Go to console.
2. Click Compute engine, and click Disks.
3. Click Create Disk.
4. Name the disk and optionally give it a description.
5. Under Location, select Regional, and choose the Region you are working in.
6. Leave Disk source type as Blank disk.
7. Under Disk type: For data volumes, select Hyperdisk Extreme. For log volume, select SSD persistent disk.
8. Under size, choose 1000GB for the data volume, and 60GB for the log volume.
9. Uncheck Enable snapshot schedule.
10. Add an appropriate label, and leave the rest of the options as default.
11. Click Create, repeating the steps to create both a data and a log volume.
12. Once the disks have been created, click Compute Engine → VM instances.
13. Click into the baseline SUT instance, and click Edit.
14. Under Storage → Additional disks, click Attach Existing Disk.
15. Choose the appropriate disk, leave Mode as Read/write, and change the Deletion rule to Delete disk.
16. Click Save.
17. Repeat steps 14 through 16 for the other disk.

Configuring Ubuntu 20.04 LTS

1. Run updates and upgrade:

```
sudo apt update
sudo apt upgrade -y
```

2. Install tools:

```
sudo apt-get install -y wget curl vim tar zip unzip nmon sysstat numactl ksh
```

3. Create directories to be mounted:

```
sudo mkdir -p /data
sudo mkdir -p /logs/log
```

4. Prepare storage:

```
sudo mkfs.xfs -f /dev/nvme0n2
sudo mkfs.xfs -f /dev/nvme0n3
```

5. Add volumes to fstab:

```
sudo vim /etc/fstab
/dev/nvme0n2 /data xfs rw,attr2,noatime 0 0
/dev/nvme0n3 /logs/log xfs rw,attr2,noatime 0 0
```

6. Mount the directories to the data and log volumes:

```
sudo mount -v /data
sudo mount -v /logs/log
```

7. Create db and backup directories:

```
sudo mkdir -p /data/db
sudo mkdir -p /data/backup
```

8. Open permissions on new folders:

```
sudo chmod -R 777 /data
sudo chmod -R 777 /logs
```

Configuring Microsoft SQL Server repository and installing SQL Server

1. Import the public repository GPG keys:

```
wget -q0- https://packages.microsoft.com/keys/microsoft.asc | sudo tee /etc/apt/trusted.gpg.d/microsoft.asc
```

2. Register the SQL Server Ubuntu repository:

```
Sudo add-apt-repository "$(wget -q0- https://packages.microsoft.com/ubuntu/20.04/mssql-server-2022.list)"
```

3. Install SQL Server:

```
sudo apt-get update
sudo apt-get install -y mssql-server
```

4. Configure the SQL installation:

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

5. For Evaluation mode, choose 1.
6. To continue, press Y.
7. Choose a password and enter it.
8. Re-enter the chosen password.
9. Ensure SQL Server is running:

```
systemctl status mssql-server --no-pager
```

Installing SQL Server command line tools

1. Enter superuser mode:

```
sudo su
```

2. Import the public repository GPG keys:

```
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
```

3. Register the Microsoft Ubuntu repository:

```
curl https://packages.microsoft.com/config/ubuntu/20.04/prod.list > /etc/apt/sources.list.d/mssql-release.list
```

4. Exit superuser mode:

```
exit
```

5. Update the sources list, and run the installation command with the unixODBC developer package:

```
sudo apt-get update
sudo apt-get install mssql-tools18 unixodbc-dev
```

6. Update to the latest version of mssql-tools:

```
sudo apt-get update
sudo apt-get install mssql-tools18
```

Configuring the Ubuntu 20.04 LTS SUT for performance

1. Increase the open file limitation on MSSQL:

```
sudo vim /etc/security/limits.d/99-mssql-server.conf
mssql hard nofile 32727
mssql soft nofile 16000
```

2. Enable Forced Unit Access (FUA) I/O subsystem capability:

```
sudo /opt/mssql/bin/mssql-conf traceflag 3979 on
sudo /opt/mssql/bin/mssql-conf set control.writethrough 1
sudo /opt/mssql/bin/mssql-conf set control.alternatewritethrough 0
```

3. Install tuneD:

```
sudo apt install tuned
```

4. Edit the MSSQL tuneD configuration file for performance. See tuned.conf for the full list of parameters.

```
sudo vim /usr/lib/tuned/mssql/tuned.conf
```


5. Make tuned.conf executable:

```
sudo chmod +x /usr/lib/tuned/mssql/tuned.conf
```

6. Enable MSSQL tuneD profile:

```
sudo tuned-adm profile mssql
```

7. Verify that the correct tuneD profile is applied:

```
sudo tuned-adm active
```

Installing the GCP CLI and downloading the database backup from Google Cloud Storage

1. Install prerequisite tools:

```
sudo apt-get install apt-transport-https ca-certificates gnupg curl sudo
```

2. Add the Google Cloud CLI distribution URL as a package source:

```
echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg] https://packages.cloud.google.com/apt cloud-sdk main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list
```

3. Import the Google Cloud public key:

```
Curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key --keyring /usr/share/keyrings/cloud.google.gpg add -
```

4. Update and install the CLI:

```
sudo apt-get update && sudo apt-get install google-cloud-cli
```

5. Initialize the CLI installation:

```
gcloud init
```

6. To choose the currently logged-in service account, press 1.
7. To choose the currently selected project, press 1.
8. To select the default zone, press Y, and choose your default zone.
9. After installation is completed, download the backup file from the previously created bucket:

```
gcloud storage cp gs://[bucket-name]/[db-backup-file-name] /data/backup/
```

Creating a machine image of your baseline SUT instance

1. Log into Google Cloud, and click Go to console.
2. Click Compute engine, and click VM instances.
3. Select the baseline SUT instance, and click More Actions (the three-dot menu on the right).
4. Click Create new machine image.
5. Give the image a name and optionally a description.
6. Under Location, select Regional, and choose the region you are working in.
7. Leave the Encryption setting at default.
8. Click Create.

Creating the client and SUT instances

To create an instance, you must first have a machine image. The steps below will walk you through the creation of an instance from a machine image.

1. Log into Google Cloud, and click Go to console.
2. Click Compute engine, and click Machine images.
3. Select your machine image, and click the right-hand More Actions menu.
4. Click Create instance.
5. Add the following information:
 - Name: Name your instance.
 - Labels: Use any appropriate labels.
 - Region: Select your desired region.
 - Zone: Select your desired zone.
 - Machine family: General-purpose
 - Series: <n2,c3>
 - Machine type: <machine type>
 - CPU platform: For n2 series: Intel Ice Lake or higher. For c3 series: Intel Sapphire Rapids or higher.
 - Click Create.
6. Repeat steps 1 through 5 for the client instance, selecting the client machine image in step 3 and choosing n2-standard-32 for the machine type in step 5.

Setting the SQL memory reserve and max degree of parallelism (MAXDOP)

1. RDP into the client instance.
2. Open the SQL Server Management Studio.
3. Right-click the SQL Instance, and click Properties.
4. Click Advanced node, and scroll down to the Max Degree of Parallelism and change the value to 0. Click OK.
5. Right-click the SQL Instance again, and go to Memory.
6. Set the Max Memory to 90% of the total memory in the system. Click OK, and close the Properties window.
7. SSH into the Ubuntu SQL VM.
8. Restart the MSSQL service:

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

Running the tests

In this section, we list the steps to run the HammerDB TPROC-C test on the instances under test. As each instance had different hardware, please refer to Table 5 to see the number of users to run on each instance.

1. On the instance you're testing, restore the database under test so that the database and log files reside on the disks under test.
2. On the client system, start HammerDB.
3. Set the database server to SQL Server, and set the workload to TPROC-C.
4. Open the Options panel for the Driver Script: SQL Server→TPROC-C→Driver Script→Options.
5. Enter the IP of the system under test in the SQL Server field.
6. Change the ODBC Driver to ODBC Driver 18 for SQL Server.

7. Choose SQL Server Authentication, and change the SQL Server User Password to the password you chose during SQL Server setup.
8. Choose Timed Driver Script.
9. Change the Total Transactions per User to 1000000000.
10. Enter 10 for the Rampup Time, and 20 for the Test Duration.
11. Click OK.
12. Open the Options panel for the Virtual Users: SQL Server→TPROC-C→Virtual User→Options.
13. Set the appropriate number of Virtual Users. We used 192 for both solutions.
14. Select the following: Show Output, Log Output to Temp, and Use Unique Log Name.
15. Click OK.
16. To capture CPU, RAM, and disk performance counters, use the following commands on the SQL instance:

```
iostat -x 2 920 > /tmp/disk_utilization
nmon -F /tmp/testmetrics.nmon -s 2 -c 920 -J -t -g auto -D
```

17. Back on the client instance, note the name of the log file. To start the run, click OK.
18. Save the HammerDB results text file and iostat/nmon output.
19. Drop and restore the database from the system under test, and reboot the system.
20. Repeat the test three times and report the median run.

Table 5: Instance configuration settings for testing. Source: Principled Technologies.

Instance	n2-highcpu-80	c3-highcpu-88
# vCPU	80	88
Memory (GB)	80	176
DB size	3,000 warehouses	3,000 warehouses
Virtual users	192	192
Warmup time	10	10
Run time	20	20

```
tuned.conf
[main]
summary=Optimize for Microsoft SQL Server
include=throughput-performance

[cpu]
force_latency=5

[sysctl]
vm.swappiness = 1
vm.dirty_background_ratio = 3
vm.dirty_ratio = 80
vm.dirty_expire_centisecs = 500
vm.dirty_writeback_centisecs = 100
vm.transparent_hugepages=always
# For multi-instance SQL deployments, use
# vm.transparent_hugepages=madvice
vm.max_map_count=1600000
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
kernel.numa_balancing=0
```

Read the report at <https://facts.pt/ft414DZ>



This project was commissioned by Intel.



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.