A Principled Technologies report: Hands-on testing. Real-world results.

A Principled Technologies report: Hands-on testing. Real-world results.

Champion big decisions and gutsy moves with the new HP Z8 Fury G5 Workstation Desktop PC

vs. an HP Z8 G4 Workstation Desktop PC

For companies with an eye to the future, equipping technical professionals with well-engineered workstations can lead to shorter analysis times for large datasets, giving data scientists the ability to run more iterations as quickly as possible. That's in part because data science is grounded in experimentation. When data scientists are working with big data or large models, the number of times they can try new configurations or explore groundbreaking parameters is limited when they don't have adequate resources to do so in a timely manner.

At Principled Technologies, we ran medical imaging, language processing, and computer vision scenarios from the MLPerf™ Inference Benchmark Suite on the new HP Z8 Fury G5 Workstation and compared its output to that of its predecessor, the HP Z8 G4 Workstation.

These results are relevant to organizations and facilities that want to advance medical research and treatment endeavors, improve customer experiences, and achieve higher levels of productivity.

Accelerate 3D medical image segmentation
More 3D U-Net samples per second

Improve customer experiences
More RNN-T and BERT-99 samples per second

Identify objects and people faster
More ResNet-50 samples per second

Champion big decisions and gutsy moves with the new HP Z8 Fury G5 Workstation Desktop PC November 2023

## The science behind the report:

# Champion big decisions and gutsy moves with the new HP Z8 Fury G5 Workstation Desktop PC

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 Champion big decisions and gutsy moves with the new HP Z8 Fury G5 Workstation Desktop PC.

We concluded our hands-on testing on September 19, 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 September 18, 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: Results of our testing. Higher samples per second are better and lower latencies are better.

| | HP Z8 Fury G5 | HP Z8 G4 | G5 win percentage |
|---|---:|---:|---:|
| 3D U-Net | | | |
| Samples per second (higher is better) | 9.98183 | 3.72338 | 168.09% |
| Mean latency (seconds) (lower is better) | 1,230.392584 | 3,301.608944 | 62.73% |
| BERT-99 | | | |
| Samples per second (higher is better) | 7,545.39 | 3,407.7 | 121.42% |
| Mean latency (seconds) (lower is better) | 404.1797261 | 427.1204625 | 5.37% |
| ResNet-50 | | | |
| Samples per second (higher is better) | 86,505.2 | 39,243.1 | 120.43% |
| Mean latency (seconds) (lower is better) | 7.023164746 | 331.0680412 | 97.88% |
| RNN-T | | | |
| Samples per second (higher is better) | 22,422.6 | 11,984.6 | 87.10% |
| Mean latency (seconds) (lower is better) | 387.205433 | 406.438205 | 4.73% |

# System configuration information

Table 2: Detailed information on the systems we tested.

| System configuration information | HP Z8 Fury G5 | HP Z8 G4 |
|---|---|---|
| Processor | | |
| Number of processors | 1 | 2 |
| Vendor | Intel® | Intel |
| Model number | Xeon® w9-3495X | Xeon 6258R |
| Core frequency (GHz) | 1.9 | 2.7 |
| Number of cores | 56 | 28 |
| Cache (MB) | 105 | 38.5 |
| Memory | | |
| Amount (GB) | 128 (8x 16) | 96 (12x 8) |
| Type | DDR5 ECC | DDR4 |
| Speed (MHz) | 4,800 | 2,666 |
| Discrete graphics | | |
| Number of cards | 4 | 2 |
| Vendor | NVIDIA® | NVIDIA |
| Model number | RTX™ 6000 Ada | RTX A6000 |
| VRAM (GB) | 48 GDDR6 | 48 GDDR6 |
| Storage | | |
| Amount (TB) | 4x 1 | 2x 1 |
| Type | PCIe®-based flash | PCIe-based flash |
| Connectivity/expansion | | |
| Wired internet | Intel Ethernet Connection (17) I219-LM | Intel Ethernet Connection (3) I219-LM |
| Wired internet | Intel I210 Gigabit Network Connection | Intel Ethernet Connection X722 |
| USB | 6 x 3.0 USB-A | 6 x 3.0 USB-A |
| Front USB | 2 x 3.0 USB-A, 2 x USB-C | 2 x 3.0 USB-A, 2 x USB-C |
| Operating system | | |
| Vendor | Windows | Windows |
| Name | 11 Pro for Workstations | 11 Pro for Workstations |
| Build number or version | 10.0.22621 Build 22621.1992 | 10.0.22621 Build 22621.1992 |

| System configuration information | HP Z8 Fury G5 | HP Z8 G4 |
|---|---|---|
| WSL2 configuration | | |
| Vendor | Ubuntu | Ubuntu |
| Name | 20.04 | 20.04 |
| Build number or version | 20.04.06 LTS | 20.04.06 LTS |
| WSL version | 2.0.0.0 | 2.0.0.0 |
| Kernel version | 5.15.123.1-1 | 5.15.123.1-1 |
| WSLg version | 1.0.57 | 1.0.57 |
| MSRDC version | 1.2.4485 | 1.2.4485 |
| BIOS | | |
| BIOS name and version | U61 Ver. 01.01.19 | HP P60 v02.91 |
| Dimensions | | |
| Height (in.) | 21.7 | 21.7 |
| Width (in.) | 8.5 | 8.5 |
| Depth (in.) | 17.5 | 17.5 |
| Weight (lb.) | 64.12 | 56.2 |

# How we tested

## Setting up the systems

### Configuring Ubuntu 20.04 on Windows 11 Windows Subsystem for Linux 2 (WSL 2)

1. In the system BIOS, confirm hardware virtualization is enabled.
2. Enable Hyper-V and Virtual Machine Platform in Windows:
    a. Open an elevated PowerShell terminal.
    b. Run the following commands, but decline the reboot until the final step:

    ```
    Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V-All
    Enable-WindowsOptionalFeature -Online -FeatureName VirtualMachinePlatform
    shutdown /r /t 0
    ```

3. Download and install drivers from NVIDIA: https://www.nvidia.com/download/index.aspx.
    a. If a professional GPU is installed, in the NVIDIA Control Panel, enable Error Correction Code.
4. To confirm GPU configuration, from an elevated terminal session, run `nvidia-smi` to list the GPUs, driver versions, and API versions, and verify that the system recognizes them.
5. Verify ECC is enabled:

    ```
    nvidia-smi -q -d ECC.
    ```

6. Install Ubuntu 20.04:

    ```
    wsl.exe --install Ubuntu-20.04.
    ```

7. Follow the prompts to create an Ubuntu user name and password, and exit the Ubuntu session.
8. Set default WSL instance to the new installation:

    ```
    wsl --set-default Ubuntu-20.04.
    ```

9. Update WSL 2 to latest release:

    ```
    wsl --update --pre-release.
    ```

10. Reboot the system:

    ```
    shutdown /r /t 0
    ```

11. Open a new Terminal session to Ubuntu 20.04.
12. Update Ubuntu:

    ```
    sudo apt update && sudo apt upgrade.
    ```

### Configuring a Collective Mind (CM) machine learning environment

1. Install prerequisites:

    ```
    sudo apt install python3 python3-pip python3-venv git wget curl zlib1g unzip.
    ```

2. Modify the /etc/profile file by adding the following lines:

```
export PATH="/home/ptuser/.local/bin:$PATH"
export PATH="/usr/local/cuda-11.8/bin:$PATH"
export LD_LIBRARY_PATH="/usr/local/cuda-11.8/lib64"
export CUDA_MODULE_LOADING=LAZY
```

3. Install CM:

```
python3 -m pip install cmind.
```

4. To add paths to the environment, exit Terminal, and restart it.
5. Test CM:

```
cm test core.
```

6. Use CM to pull the MLCommons GitHub repository:

```
cm pull repo mlcommons@ck.
```

### Configuring WSL NVIDIA support

1. Install CUDA for Linux:

```
mkdir nvidia-prereq
cd nvidia-prereq
wget https://developer.download.nvidia.com/compute/cuda/11.8.0/local_installers/
cuda_11.8.0_520.61.05_linux.run
sudo sh cuda_11.8.0_520.61.05_linux.run
```

2. Test CUDA support:

```
cmr "get cuda-devices"
```

3. Install CUDA into CM:

```
cmr "get cuda"
```

4. Install cuDNN & TensorRT:

```
cmr "get cudnn" --tar_file=~/nvidia-prereq/cudnn-linux-x86_64-8.9.5.29_cuda11-archive.tar.xz
cmr "get tensorrt _dev" --tar_file=~/nvidia-prereq/TensorRT-8.6.1.6.Linux.x86_64-gnu.cuda-11.8.tar.gz
```

5. Install system dependencies:

```
cm run script "get sys-utils-cm" --quiet
cm run script "get python" --version_min=3.8
pip install packaging tqdm
```

### Confirming the CM environment recognizes the GPUs

1. Run `cmr get-cuda-devices`
2. Confirm the devices listed match the system configuration.

Note: At the time of testing, there was a bug with WSL 2 and NVIDIA driver integration for systems with more than two GPUs that required toggling the developer option “Manage GPU Performance Counters,” which would reset the driver/WSL 2 integration and allow the command in step 1 to run successfully.

## Running ML testing scripts

We ran scripts to execute the machine learning workloads and test performance. We provide those scripts below.

### 3d-unet-99

```
cmr "generate-run-cmds inference _find-performance" \
    --model=3d-unet-99 \
    --implementation=nvidia-original \
    --device=cuda \
    --backend=tensorrt \
    --category=edge \
    --division=open \
    --execution-mode=valid \
    --results_dir=$HOME/MLPerf_OOB \
    --quiet \
    --clean
```

#### bert-99

```
cmr "generate-run-cmds inference _find-performance" \
    --model=bert-99 \
    --implementation=nvidia-original \
    --device=cuda \
    --backend=tensorrt \
    --category=edge \
    --division=open \
    --execution-mode=valid \
    --results_dir=$HOME/MLPerf_OOB \
    --quiet \
    --clean
```

#### resnet-50

```
cmr "generate-run-cmds inference _find-performance" \
    --model=resnet50 \
    --implementation=nvidia-original \
    --device=cuda \
    --backend=tensorrt \
    --category=edge \
    --division=open \
    --execution-mode=valid \
    --results_dir=$HOME/MLPerf_OOB \
    --quiet \
    --clean
```

### rnnt

At the time of the testing, there was not an existing profile for the Ada architecture GPUs, so MLCommons developers suggested using the L4 preset.

```
cmr "generate-run-cmds inference _find-performance" \
    --model=rnnt \
    --implementation=nvidia-original \
    --device=cuda \
    --backend=tensorrt \
    --category=edge \
    --division=open \
    --execution-mode=valid \
    --results_dir=$HOME/MLPerf_OOB \
    --quiet \
    --clean \
    --gpu_name=l4 (for the Ada GPUs only)
```

Read the report at https://facts.pt/WxGpr9S ▶

This project was commissioned by HP.

Principled Technologies®

Facts matter.®

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.