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

Get results from demanding workflows in less time with the new HP Z8 Fury G5 Workstation Desktop PC

Compared to a Lenovo ThinkStation P620 Tower Workstation

Today's creatives and technical professionals face a common challenge—demanding workflows that require significant processing power. And, as always, time is money.

At Principled Technologies, we compared CPU and GPU performance using benchmarks and several MLPerf™ machine learning (ML) model performance between an HP Z8 Fury G5 Workstation Desktop PC and a Lenovo® ThinkStation® P620 Tower Workstation. We equipped both workstations with 128 GB of RAM (plenty for our tests) and configured them with the most powerful CPU and GPU hardware available at the time of testing. We loaded the HP Z8 Fury G5 with an Intel® Xeon® w9-3495X CPU and four NVIDIA® RTX™ 6000 Ada Generation GPUs. And we loaded the Lenovo ThinkStation P620 with an AMD Ryzen™ Threadripper PRO 5995WX CPU and two NVIDIA RTX A6000 GPUs with NVLink.

These results are relevant to creatives and technical professionals who do such computationally intense work as generating photorealistic images; running advanced simulations and visualizations; depending on hardware-intensive computer-aided design (CAD); or working with complex datasets that include videos, images, and speech.

Accelerate 3D modeling and rendering
Higher Cinebench R23, Geekbench 6 Pro, Blender 3.5, and Maxon Redshift benchmark scores

Expedite project cycles
Faster Blender 3.6 renders while processing more samples per minute

Tackle complex AI/ML problems
More 3D U-Net, BERT-99, RNN-T, and ResNet-50 samples per second

Get results from demanding workflows in less time with the new HP Z8 Fury G5 Workstation Desktop PC November 2023

The science behind the report:

Get results from demanding workflows in less time 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 Get results from demanding workflows in less time 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 performance testing. Higher benchmark score are better and lower times are better.

	HP Z8 Fury G5	Lenovo® ThinkStation® P620	G5 win percentage
Cinebench R23 vR23.200 benchmark scores (higher is better)			
Median CPU multi-core score	60,194	68,371	-11.96%
Median CPU single-core score	1,669	1,485	12.39%
Geekbench 6 Pro v6.0.3 benchmark (higher is better)			
Median CPU multi-core score	17,575	20,016	-12.20%
Median CPU single-core score	2,168	2,044	6.07%
Median GPU Compute OpenCL (single-GPU) score	279,364	191,089	46.20%
Blender v3.5.0 GPU rendered samples per minute (higher is better)			
Monster	6,848.09	2,910.87	135.25%
Junkshop	3,194.20	1,744.34	83.11%
Classroom	3,212.68	1,493.47	115.11%

	HP Z8 Fury G5	Lenovo® ThinkStation® P620	G5 win percentage
Blender v3.6.1 multi-GPU Splash Rendering with 200% resolution, 256 samples (2,304 total samples). Times in mm:ss.00 (lower is better). Higher samples per minute is better.			
Median CUDA render time	00:44.17	01:36.05	54.01%
Median CUDA samples per minute	3129.73	1,439.25	117.46%
Median OptiX render time	00:51.72	01:07.59	23.48%
Median OptiX samples per minute	2,672.85	2,045.27	30.68%
PugetBench for Lightroom Classic v0.94 (using Adobe Lightroom® Classic v12.4) benchmark scores (higher is better)			
Median overall score	1,510.5	1,195	26.40%
Maxon Redshift v3.5.17 render time in mm:ss (lower is better).			
Median render time	00:41	01:25	51.76%

Table 2: Results of our data science testing. Higher samples per second are better and lower latencies are better.

	HP Z8 Fury G5	Lenovo ThinkStation P620	G5 win percentage
3D U-Net			
Samples per second (higher is better)	9.98183	3.70866	169.15%
Mean latency (seconds) (lower is better)	1,230.392584	3,311.985679	62.85%
BERT-99			
Samples per second (higher is better)	7,545.39	3,422.93	120.44%
Mean latency (seconds) (lower is better)	404.1797261	403.6212782	-0.14%
ResNet-50			
Samples per second (higher is better)	86,505.2	39,870.9	116.96%
Mean latency (seconds) (lower is better)	7.023164746	331.2324044	97.88%
RNN-T			
Samples per second (higher is better)	22,422.6	13,031.1	72.07%
Mean latency (seconds) (lower is better)	387.205433	412.918578	6.23%

System configuration information

Table 3: Detailed information on the systems we tested.

System configuration information	HP Z8 Fury G5	Lenovo ThinkStation P620
Processor		
Number of processors	1	1
Vendor	Intel®	AMD
Model number	Xeon® w9-3495X	Ryzen™ Threadripper PRO 5995WX
Core frequency (GHz)	1.9	2.7
Number of cores	56	64
Cache (MB)	105	256
Memory		
Amount (GB)	128 (8x 16)	128 (8x 16)
Type	DDR5 ECC	DDR4
Speed (MHz)	4,800	3,200
Discrete graphics		
Number of cards	4	2
Vendor	NVIDIA®	NVIDIA
Model number	RTX™ 6000 Ada	RTX A6000
VRAM (GB)	48 GDDR6	48 GDDR6
Multi-GPU Interconnect	N/A	NVLink
Storage		
Amount (TB)	4x 1	1
Type	PCIe®-based flash	PCIe-based flash
Connectivity/expansion		
Wired internet	Intel Ethernet Connection (17) I219-LM	Marvell AQtion 10Gbit Network Adapter
Wired internet	Intel I210 Gigabit Network Connection	N/A
USB	6x 3.0 USB-A	6x 3.0
Front USB	2x 3.0 USB-A, 2x USB-C	2x 3.0, 2x USB-C
Operating system		
Vendor	Windows	Windows
Name	11 Pro for Workstations	11 Pro
Build number or version	10.0.22621 Build 22621.1992	10.0.22621 Build 22621.1992

System configuration information	HP Z8 Fury G5	Lenovo ThinkStation P620
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	Lenovo S07KT51A
Dimensions		
Height (in.)	21.7	17.3
Width (in.)	8.5	6.5
Depth (in.)	17.5	18.1
Weight (lb.)	64.12	33.76

How we tested

Setting up the systems to test performance with benchmarks

Setting up and updating the OEM image

1. Boot the system.
2. To complete installation, follow the on-screen instructions, using the default selections when appropriate.
3. Set the Windows Power Plan to Ultimate Performance.
4. Set Screen and Sleep options to Never:
    a. Right-click the desktop, and select Display settings.
    b. Select System from the left-hand column.
    c. Click Power.
    d. For all power options listed under Screen and Sleep, select Never.
5. Disable User Account Control notifications:
    a. Select Windows Start, type `UAC`, and press the Enter key.
    b. Move the slider control to Never notify, and click OK.
6. Run Windows Update, and install all updates available.
7. Launch each vendor proprietary utility app installed on each system, and update any drivers or BIOS files:
    a. For Lenovo, run the Lenovo System Update utility.
    b. For HP, run the HP Support Assistant utility.
8. Boot to system BIOS and select the following settings:
    a. For Lenovo, click Power Settings→Fan Control Stepping, and set it to 7 (higher).
    b. For HP, click Advanced→Performance Management→Performance Control, and set it to High Performance.
9. Download and install the latest NVIDIA Drivers from https://www.nvidia.com/download/index.aspx.
10. Verify the date and time are correct, and synchronize the system clock with the time server.
11. Pause Automatic Windows Updates:
    a. Click Windows Start.
    b. Type Windows Update settings, and press Enter.
    c. From the Pause updates drop-down menu, select Pause for 5 weeks.

Capturing an image

1. Connect an external HDD to the system.
2. Click Windows Menu, and type `Control Panel` in the search bar. Click Control Panel→System, Security→Backup and Restore (Windows 7)→Create a system image.
3. Verify that the external HDD is selected as the save drive, and click Next.
4. Verify that all drives are selected to back up, and click Next.
5. Click Start backup.
6. When asked if you want to create a system repair disc, select No, and close the dialogs.

Restoring an image

1. Connect an external HDD to the system.
2. Press and hold the Shift key while restarting the system.
3. Select Troubleshoot.
4. Select Advanced options.
5. Select See more recovery options.
6. Select System image recovery.
7. Select the User account.
8. Enter the system password, and click Continue.
9. At the Restore system files and settings screen, select Next.
10. Verify that the external HDD is selected, and click Next.
11. Once the recovery has completed, click Finish.

Testing performance with Cinebench R23

Setting up the test

1. Download and install Cinebench from https://www.maxon.net/en/downloads/cinebench-r23-downloads.

Running the test

1. Launch Cinebench.
2. Select File→Advanced benchmark.
3. Set the Minimum Test Duration to Off.
4. Select either CPU (Multi Core) or CPU (Single Core), and click Start
5. Record the result.
6. Wait 15 minutes before re-running.
7. Repeat steps 1 through 6 two more times, and record the median result.

Testing performance with Geekbench 6 Pro

Setting up the test

1. Purchase a Pro license, and download and install Geekbench 6 Pro from https://www.geekbench.com/download/.

Running the test

1. Launch Geekbench.
2. Click Run CPU Benchmark.
3. Record the result.
4. Wait 15 minutes before re-running.
5. Repeat steps 1 through 4 two more times, and record the median result.

Running the GPU OpenCL test

1. Launch Geekbench.
2. Click Run GPU Compute Benchmark.
3. Record the result.
4. Wait 15 minutes before re-running.
5. Repeat steps 1 through 4 two more times, and record the median result.

Testing performance with Blender

Setting up the test

1. Download the Blender benchmark from https://opendata.blender.org/.

Running the test

1. Launch the Blender benchmark.
2. At the Welcome screen, click Next.
3. Select Blender version 3.5.0, and click Next.
4. At the Benchmark Scenes screen, click Next.
5. At the Benchmark Device screen, select the GPU option, and click Start Benchmark.
6. Record the results.
7. Wait 15 minutes before performing the next run.
8. Repeat steps 1 through 7 two more times, and record the median result.

Testing performance with Blender custom multi-GPU rendering

Setting up the test

1. Download and install Blender from https://www.blender.org/download/.
2. Download the demo scene from https://download.blender.org/demo/splash/blender-3.5-splash.blend.

Running the test with CUDA

1. Launch Blender.
2. At the splash screen, click Open, and open the demo scene.
3. Click Edit→Preferences→System.
4. Select the CUDA tab, check the boxes for all GPUs to be tested, and close the window.
5. In the right-side Render Properties pane, under Device, select GPU Compute.
6. In the right-side Output Properties pane, set the Resolution percentage to 200%.
7. To add a new workspace, next to the Main workspace, click +, and select General→Rendering.
8. In the right-side Render Properties pane, set the Render Max Samples and Min Samples to 256.
9. Click Render→Render Image.
10. Record the results.
11. Wait 15 minutes before performing the next run.
12. Repeat steps 1 through 11 two more times, and record the median result.

Running the test with OptiX

1. Launch Blender.
2. At the splash screen, click Open, and open the demo scene.
3. Click Edit→Preferences→System.
4. Select the OptiX tab, check the boxes for all GPUs to be tested, and close the window.
5. In the right-side Render Properties pane, under Device, select GPU Compute.
6. In the right-side Output Properties pane, set the Resolution percentage to 200%.
7. To add a new workspace, next to the Main workspace, click +, and select General→Rendering.
8. In the right-side Render Properties pane, set the Render Max Samples and Min Samples to 256.
9. Click Render→Render Image.
10. Record the results.
11. Wait 15 minutes before performing the next run.
12. Repeat steps 1 through 11 two more times, and record the median result.

Testing performance with Maxon Redshift

Setting up the test

1. Download and install the Maxon app from https://www.maxon.net/en/try.
2. Launch the Maxon app.
3. Click the Maxon One 14-day trial, and download and install the Redshift application.

Running the test

1. To open a command prompt, click the Windows icon, type `cmd`, and press Enter.
2. Inside the command prompt, `type cd C:\ProgramData\Redshift\bin`, and press Enter.
3. To start the benchmark, type `RunBunchmark.bat`, and press Enter.
4. Record the results.
5. Wait 5 minutes before rerunning the test.
6. Repeat steps 1 through 5 two more times, and record the median result.

Testing performance with PugetBench for Lightroom Classic

Setting up the test

1. Launch Adobe Lightroom Classic.
2. To disable tips, on the popup tip, click the checkbox to turn off tips.
3. Close Adobe Lightroom Classic.
4. Purchase and download the PugetBench for Lightroom Classic license from https://www.pugetsystems.com/labs/articles/PugetBench-for-Adobe-Creative-Cloud-1642/.
5. Download and extract the plugin and assets from https://www.pugetsystems.com/labs/articles/PugetBench-for-Adobe-Creative-Cloud-1642/.
6. Open Adobe Lightroom Classic.
7. Click File→Plug-in Manager, and click Add.

8. Navigate to the location of the benchmark folder, select the pugetsystems.lrplugin folder, and click Done.
9. Click File→Plug-in Extras→Benchmark Run.
10. Enter your license key, and click Save/Update Settings.

Running the test

1. Boot the system.
2. Open Adobe Lightroom Classic.
3. Click File→Open Catalog, and open the Benchmark Catalog.lrcat file.
4. Click Relaunch.
5. Click File→Plug-in Extras→Benchmark Run.
6. Click Run Benchmark.
7. When the benchmark finishes, record the overall score.
8. Close Adobe Lightroom Classic, and restart the system under test.
9. Wait 15 minutes before performing the next run.
10. Repeat steps 1 through 9 two more times, and record the median result.

Setting up the systems to run machine learning workloads

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 machine learning 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/b86SBHo ▶

This project was commissioned by HP.

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.