NVIDIA GH200 Grace Hopper Superchip Benchmark Step-by-Step Guide

Application Note

DA-11356-002_06 | September 2024

Document History

DA-11356-002_06

Version	Date	Description of Change
01	June 6, 2023	Initial release
02	October 2, 2023	> Updated the “Introduction” section. > Updated the “GPU STREAM” section. > Added the “CPU STREAM” section. > Updated the sustained GEMM. > Added CuFFT and the attachment.
03	October 17, 2023	> Added the DALI app. > Removed Zero copy GEMM.
04	February 15, 2024	Updated scripts attachment.
05	July 16, 2024	Removed the following attachments from the PDF file: > `stream_test.nv7z` > `cublasMatmulBench.nv7z`
06	September 5, 2024	> Added HPL Performance and the attachment. > Added HPL- MxP Performance and the attachment. > Added Llama-3 8B inference performance and the attachment. > Removed the `cufftBench.nv7z` attachment from the PDF file.

Table of Contents

List of Tables

Introduction

This application note compares NVIDIA® GH200 benchmark data to the NVIDIA DGX™ H100 platform. This initial version of the application note provides benchmarks for low-level performance metrics for bandwidth and throughput. However, this application note will be updated over time to include more workloads and application performance data.

The NVIDIA GH200 Grace Hopper™ Superchip architecture combines the groundbreaking performance of the following:

- NVIDIA Hopper™ GPU with the versatility of the NVIDIA Grace™ CPU, connected with a high bandwidth.
- A memory coherent NVIDIA NVLink™ Chip-2-Chip (C2C) interconnect in a superchip.
- Support for the new NVLink Switch System.

The GH200 system is set with Ubuntu 22.04, NVIDIA CUDA® 12.3, and NVIDIA Driver 545.14. The GH200 benchmark numbers in this application note are preliminary and subject to change.

DGX H100 benchmark numbers were measured on an Intel® Xeon® Platinum 8480C system. The clocks were set to the maximum at 1,980 MHz for GPU and 2,619 MHz for GPU memory with ECC enabled on Ubuntu 22.04, CUDA 12, and NVIDIA Driver 525.85.

The H100 SXM5 80 GB benchmark numbers are preliminary and are only presented for comparisons to GH200.

Partner benchmark results will vary based on a variety of factors such as ambient temperature, hardware, software, thermal design, and server configurations. These benchmark numbers are meant only as a reference data point.

Note: Run-to-run variation up to 3% in delivered performance on the same system is considered normal.

Important: All benchmark numbers are preliminary and represent performance at the launch of the GH200 Grace Hopper Superchip and will be updated after the products become generally available. The CUDA and NVIDIA driver software stack with Deep Learning (DL) frameworks and applications are continuously updated, so the performance will vary over time.

Refer to the following pages for the latest DL and high-performance computing (HPC) performance results:

- https://developer.nvidia.com/deep-learning-performance-training-inference
- https://developer.nvidia.com/hpc-application-performance).

Table 1. System Specifications

System Specification	NVIDIA DGX H100	NVIDIA GH200 Grace Hopper Superchip
GPU	8x NVIDIA H100 80 GB	1x NVIDIA H100 96 GB
CPU	Dual Intel Xeon Platinum 8480C, 2 GHz, 56 cores	Grace CPU, 3.1 GHz, 72 cores
System memory	2 TB DDR5	120 GB LPDDR5X 480 GB LPDDR5X

Libraries and Benchmarks

NVIDIA CUDA-X™, which is built on CUDA, is a collection of libraries, tools, and technologies that deliver dramatically higher performance compared to CPU-only alternatives across multiple application domains from artificial intelligence (AI) to HPC.

There are also many CUDA code samples in the CUDA toolkit. To highlight GH200, the following sections provide information about a few options.

GPU STREAM

NVIDIA provides an optimized CUDA implementation for the STREAM benchmark to measure memory bandwidth on a Hopper GPU. In addition to the four kernels in STREAM, this implementation also includes basic load and store tests to measure read and write memory bandwidth.

Usage

To run STREAM, use the `hpc-benchmarks:24.03` container from https://catalog.ngc.nvidia.com/orgs/nvidia/containers/hpc-benchmarks.

```
-n<elements>: number of double precision-floating point elements
```

Command

```
$ ./stream_test -n1308622848
```

Interpreting the results

H100 SXM5 80GB has 80 GB of HBM3 with peak memory bandwidth of 3,352 GB/s, and GH200 Hopper GPU has 96 GB of HBM3 with peak memory bandwidth of 4,023 GB/s.

Table 2. GPU STREAM Benchmark

	GPU Memory Bandwidth (GB/s)	
STREAM	DGX H100 80GB	GH200 96 GB
Copy	3067	3666
Scale	3060	3667
Add	3128	3754
Triad	3132	3755

CPU STREAM

The STREAM benchmark is a simple, synthetic benchmark program that measures sustainable main memory bandwidth in MB/s and the corresponding computation rate for simple vector kernels on a CPU.

Usage

To run STREAM, use the `hpc-benchmarks:24.06` container from https://catalog.ngc.nvidia.com/orgs/nvidia/containers/hpc-benchmarks.

```
-n<elements>: number of double precision-floating point elements
```

Commands

```
stream-test-cpu.sh -n 1308622848
```

Interpreting the results

- The GH200 Grace CPU has 120 GB of LPDDR5X with a peak memory bandwidth of 512 GB/s.
- The GH200 Grace CPU has 480 GB of LPDDR5X with a peak memory bandwidth of 384 GB/s.

Table 3. CPU STREAM Benchmark

STREAM	CPU Memory Bandwidth (GB/s)	
	GH200 120 GB	GH200 480 GB
Copy	448	342
Scale	448	345
Add	442	336
Triad	444	340

NVBandwidth

This tool measures the bandwidth on GPUs.

Commands

To run the test, run the following commands.

```
$ git clone https://github.com/NVIDIA/nvbandwidth
$ sudo ./debian_install.sh
$ cmake .
$ make
$ ./nvbandwidth
```

Interpreting the results

NVLink-C2C is an NVIDIA memory coherent, high-bandwidth, and low-latency superchip interconnect that delivers up to 900 GB/s total of bidirectional bandwidth.

When you look at host_to_device_memcpy_sm and device_to_host_memcpy_sm, each row represents the measured single directional bandwidth between the host and the device for a GPU.

Table 4. NVBandwidth

NVBandwidth	NVLink-C2C Bandwidth (GB/s)
	GH200
Host to device	419
Device to host	371

High Performance LINPACK

NVIDIA has a GPU-accelerated implementation of High Performance LINPACK (HPL, which primarily stress tests the system's FP64 throughput.

To run this benchmark, download the HPL-MxP.txt file, which is in the HPL_HPL-MxP benchmarks.nv7z file that is attached to this PDF.

Command Line

```
nvidia-smi --reset-gpu-clocks

docker run --runtime nvidia --shm-size=1g --ulimit memlock=-1 --privileged -e
USER_ID=$(id -u) -e USER_NAME=$(id -un) -e GROUP_ID=$(id -g) -e GROUP_NAME=$(id -gn) -it
"nvcr.io/nvidia/hpc-benchmarks:24.06" mpirun --allow-run-as-root -np 1 --mca pml ucx --
mca btl ^openib,smcuda  -mca coll_hcoll_enable 0 -x coll_hcoll_np=0 --bind-to
none  ./hpl.sh --dat hpl-linux-aarch64-gpu/sample-dat/HPL-GH-1GPU.dat
```

Interpreting Results

HPL was measured using the nvcr.io/nvidia/hpc-benchmarks:24.06 container image. HPL solves the Ax=B liner system of equations, and HPL performance is bounded by DGEMM performance. The performance difference between HPL and DGEMM is due to different matrix sizes, input coefficients, type of initialization, and time of execution.

GH200 shows higher performance for the following reasons:

- Larger GPU memory, which enables larger problem sizes to be run.
- Higher TDP on GH200 and dynamic power sharing between the CPU and the GPU.
- Higher CPU core count per GPU.

Table 5. HPL Performance

Number of GPUs	HPL Performance (TFLOPs)	
	DGX H100 80GB	GH200 96GB
1	47	52

High Performance LINPACK – Mixed Precision

The HPL-MxP benchmark highlights the emerging convergence of HPC and artificial intelligence (AI) workloads.

NVIDIA also has a GPU acceleration implementation of High Performance LINPACK – Mixed Precision (HPL-MxP) that uses mixed-precision iterative and direct methods to utilize mixed-precision tensor cores. To run this benchmark, download the `HPL-MxP.txt` file, which is in the `HPL_HPL-MxP benchmarks.nv7z` file that is attached to this PDF.

Command Line

```
nvidia-smi --reset-gpu-clocks

docker run --runtime nvidia --shm-size=1g --ulimit memlock=-1 --privileged -e
USER_ID=$(id -u) -e USER_NAME=$(id -un) -e GROUP_ID=$(id -g) -e GROUP_NAME=$(id -gn) -it
"nvcr.io/nvidia/hpc-benchmarks:24.06" mpirun --allow-run-as-root -np 1 --mca pml ucx --
mca btl ^openib,smcuda -mca coll_hcoll_enable 0 -x coll_hcoll_np=0 --bind-to none bash -c
"OMP_NUM_THREADS=72 ./hpl-mxp.sh --n 148880 --nb 4096 --nprow 1 --npcol 1 --nporder row -
-gpu-affinity 0 --cpu-affinity 0-71 --mem-affinity 0 --ucx-affinity
mlx5_0:mlx5_1:mlx5_3:mlx5_4:mlx5_7:mlx5_8:mlx5_9:mlx5_10 --preset-gemm-kernel 0 --u-
panel-chunk-nbs 16 --use-mpi-panel-broadcast 80 --call-dgemv-with-multiple-threads 0 --
Anq-device 0 --mpi-use-mpi 1 --prioritize-trsm 0 --prioritize-factorization 1 --sloppy-
type 1"
```

Interpreting Results

HPL-MxP was measured using the `nvcr.io/nvidia/hpc-benchmarks:24.06` container image. HPL-MxP solves the Ax=B liner system of equations with LU factorization in FP32, which uses FP16, FP8 GEMM, or GMRES solver in FP64 internally.

> **Note:** The measurements in Table 6 are for FP8.

Like HPL, the performance difference between HPL-MxP and FP8 GEMM is due to different matrix sizes, input coefficients, type of initialization, and time of execution. GMRES is also a bandwidth bound operation, which by default runs on CPUs that significantly affect the performance.

GH200 shows higher performance for the following reasons:

- Larger GPU memory, which enables larger problem sizes to be run.
- Higher TDP on GH200 and dynamic power sharing between CPUs and GPUs.
- Higher CPU core count per GPU.
- CPU-GPU communication is faster on GH200.

Table 6. HPL-MxP Performance

Number of GPUs	HPL-MxP Performance (TFLOPs)

	DGX H100	GH200
1	307	430

Attachments

The following files are attached to this application note:

- HPL_HPL-MxP benchmarks.nv7z
- Scripts_for_apps_v3.nv7z
- DL_inference_scripts.nv7z

To access the attached files, click the **Attachment** icon on the left-hand toolbar on this PDF (using Adobe Acrobat Reader or Adobe Acrobat). Select the file and use the Tool Bar options (**Open, Save**) to retrieve the documents. Files with the .nv7z extension must be renamed to .7z and extracted using the 7-Zip file archive software.

Application Performance

NVIDIA DALI for ResNet50

To run this benchmark, download the `scripts_for_apps_v3.nv7z` file that is attached to this PDF.

NVIDIA Data Loading Library (DALI) is a portable, open-source library that is used to decode and augment images, videos, and speech to accelerate DL applications. DALI reduces latency and training time, which mitigates bottlenecks by overlapping training and preprocessing. It provides a drop-in replacement for built-in data loaders and data iterators in popular DL frameworks for easy integration or retargeting to different frameworks.

Interpreting the results

DALI (v1.30) for ResNet50, using a Hopper GPU, was run on DGX H100 and GH200 systems. Faster data access to the CPU memory through NVLink-C2C and a higher CPU and GPU ratio with GH200 provides boosts the data processing performance by 1.5x.

Table 7. DALI for ResNet50

DALI for RN50	Images/s	
	DGX H100	**GH200**
Typical ResNet50 data processing pipeline running on ImageNet like JPEG test data set (VGA, WXGA, HD). Image decoding->random resized crop->normalization and random flip to 224x224, NCHW format, FP16	19,885	29,757

Llama-3 8B

- For the source code for NVIDIA TensorRT™-LLM, go to https://github.com/NVIDIA/TensorRT-LLM.
- To run this benchmark, download the `DL_inference_scripts.txt` and the config files in the `DL_inference_scripts.nv7z` file that is attached to this PDF.

Table 8. GH200 Llama-3 8B Inference at Latency Requirement

Llama-3 8B Inference	GH200	H100

	First Token Latency (ms)	Throughput (Tokens/sec)	First Token Latency (ms)	Throughput (Tokens/sec)
BS=8, ISL/OSL=2048/128, TP=1, PP=1 (2s first token latency requirement)	209	1114	220	1030
BS=64, ISL/OSL= 128/128, TP=1, PP=1 (1s first token latency requirement)	101	9895	107	8732
BS=64, ISL/OSL= 128/2048, TP=1, PP=1 (1s first token latency requirement)	126	6981	1112	6424
BS=4, ISL/OSL= 2048/2048, TP=1, PP=1 (1s first token latency requirement)	108	752	114	681

Interpreting Results

Llama-3 8B inference is measured on FP8 precision on 1x GH200 96GB and 1x H100 80GB. Table 8 includes the throughput and first token latency for Llama-3 8B model.