



Facilitating Research on the NVIDIA Grace Hopper Superchip: The CU Boulder Research Computing Experience

Facilitating Research on the NVIDIA Grace Hopper Superchip: The CU Boulder Research Computing Experience

Presenter: Brandon Reyes

- Website: www.rc.colorado.edu
- Helpdesk: rc-help@colorado.edu
- Slides:
https://github.com/ResearchComputing/rmacc_2025



Session Overview

- GH200 architecture overview
- GH200 software stack
- Beta testing phase
- Successful use cases
- Potential future directions
- How can you run on the GH200s?

GH200 architecture overview

The Grace Hopper Superchip (GH200) is a newer chip provided by NVIDIA. Its unique architecture allows the GPU and CPU to efficiently share and exchange memory.

- Ideal for workloads that utilize the CPU and GPU to a high degree and require a large amount of memory
- Although powerful, the Arm-based CPU can present support challenges
 - HPC systems often have software built only for x86_64 architectures
 - Utilizing the full capabilities can require CUDA code changes

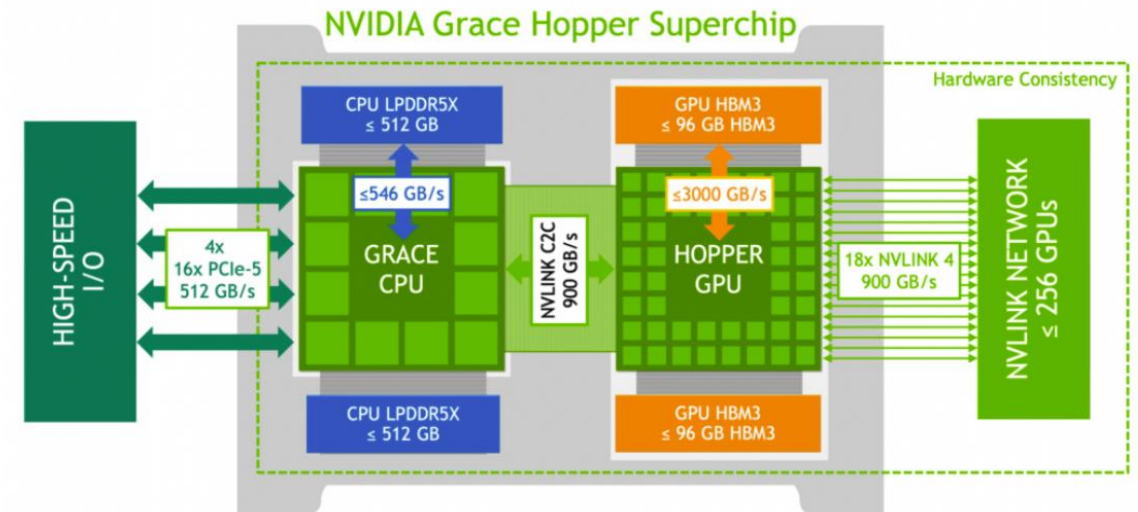


Image provided by <https://resources.nvidia.com/en-us-grace-cpu/nvidia-grace-hopper>

Specifications for our 2 GH200 nodes

- Grace CPU has 72 cores and roughly 480 GB of RAM
 - Arm based (Neoverse V2)
- Hopper GPU has roughly 100 GB of VRAM
- CPU and GPU are connected via NVIDIA NVLink-C2C
 - This enables efficient and seamless memory transfer between the two components
- FAST I/O speed (512 GB/s)
 - Top I/O speeds only apply to the SSD
- Roughly 1.7 TB of usable SSD on the node

GH200 software stack

Due to the unique architecture and Arm-based processors, we opted for a curated and paired-down software stack.

- CUDA compilers through NVIDIA HPC SDK e.g. nvcc, nvc++
- CUDNN libraries
- Miniforge (mamba and conda)
 - We found that most libraries that provided an aarch64 version (with GPU capabilities) worked on the GH200s
- Apptainer
 - A large selection of compatible containers are available through NVIDIA's NGC catalog

Beta testing phase

Goal: to identify any issues associated with the GH200s, software requirements, and workflows that could take advantage of the architecture

- Directed towards users with established GPU workflows
- An initial consultation was held to determine if their existing GPU workflows met the following criteria:
 - Peak GPU usage was roughly half the size of the VRAM
 - Peak CPU utilization was at least 70%
 - Exceptions were made for applications that required the large amount of available RAM
 - Software utilized was available for the GH200s

Beta testing phase cont.

Once approved, we

- Installed all necessary software and ran a trimmed-down version of the user's code
- Added users to the provided reservation
 - Enabled us to use established QoS and provide access only to approved users
- Created an allocation specific to the user
 - Facilitated easier analysis of hardware usage
- Onboarded the user to the node
 - Provided them information on how to run on the node and how to install their own software
- Provided hands-on support for any issues encountered

Successful use cases

All users in the beta testing phase reported that they would not be able to run their workflows without the GH200s!

- All user workflows were AI based:
 - CNNs, LSTMs, KANs, MLPs
 - Gradient boosted trees (XGBoost)
 - Image segmentation (SAM2)
- Training large models and hyperparameter tuning
 - We have seen at least 2X speedup, some users reported 10X speedup
- Inference for Large Language Models
 - We were able to run Llama 3.1 405b (requires ~300 GB of memory)

Potential future directions

- Moving out of the beta testing phase:
 - Node access will be provided via a QoS
 - Job submissions will be limited to 1 per user
 - Workflows will continue to be evaluated before permission is granted
 - Ensures proper node use and well-informed users
- We are considering utilizing Multi-Instance GPU (MIG) on one of the nodes
 - MIG would provide more resources for prototyping and testing purposes
 - We found that some users could continue to benefit from the GH200 architecture, even if they had half the GPU resources

How can you run on the GH200s?

- Submit a ticket to rc-help@colorado.edu
- In the subject provide: I am interested in running “X application” on the GH200s
- In the email body:
 - Short description of what your workflow does
 - Why you believe the GH200s would be beneficial to your workflow
 - Provide us with Linux paths to the code you would like to run
 - If possible, any JobIDs of this workflow you have ran on a GPU node

Thank you!

Slides:

https://github.com/ResearchComputing/rmacc_2025

