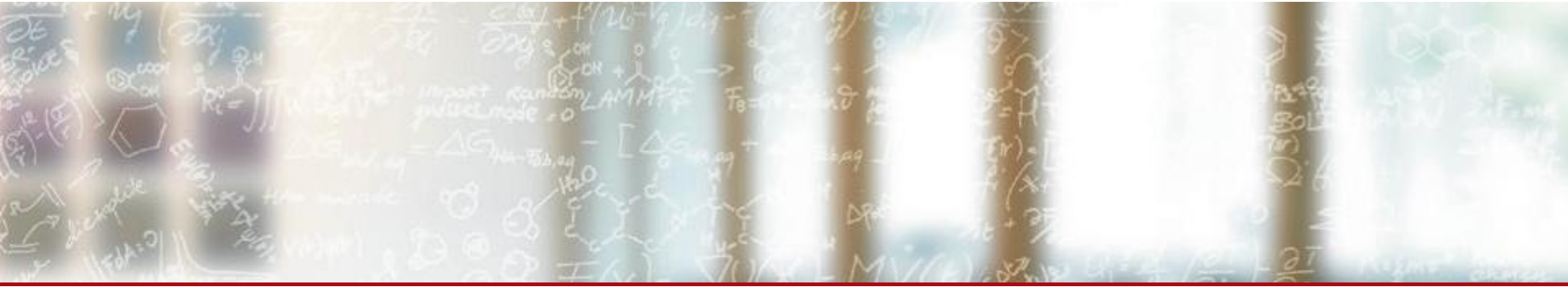


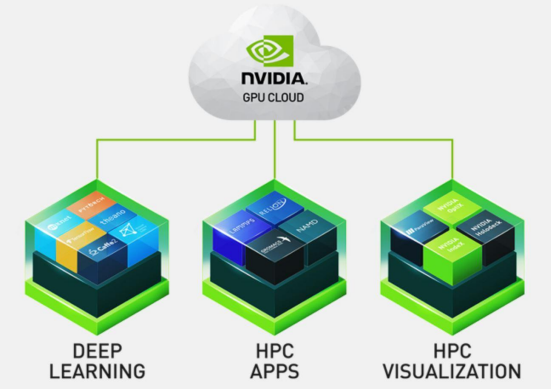
Enabling Performance for NGC Containers on the Slingshot 11 Interconnect

CANOPIE-HPC Workshop @ SC23
Alberto Madonna, ETH Zurich / CSCS

November 13th, 2023

NVIDIA GPU Cloud (NGC) images

- *"The NGC Catalog is a curated set of GPU-optimized software for AI, HPC and Visualization."*



- NGC container images are widely adopted to deploy ML/AI frameworks and workflows

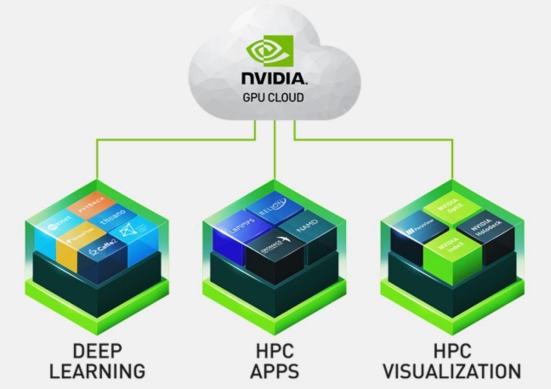
HPE Slingshot 11

- Latest generation high-performance interconnect for HPE systems
- Used in several world-class supercomputers

- Ethernet compatible, innovative congestion management, adaptive routing, QoS

NVIDIA GPU Cloud (NGC) images

- *"The NGC Catalog is a curated set of GPU-optimized software for AI, HPC and Visualization."*



- NGC container images are widely adopted to deploy ML/AI frameworks and workflows

- Built on OpenMPI+UCX, no libfabric

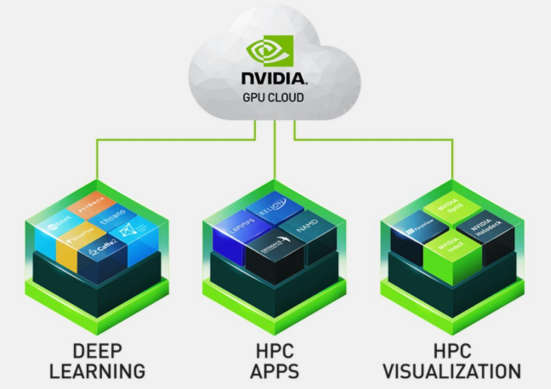
HPE Slingshot 11

- Latest generation high-performance interconnect for HPE systems
- Used in several world-class supercomputers

- Ethernet compatible, innovative congestion management, adaptive routing, QoS

NVIDIA GPU Cloud (NGC) images

- *"The NGC Catalog is a curated set of GPU-optimized software for AI, HPC and Visualization."*



- NGC container images are widely adopted to deploy ML/AI frameworks and workflows
- Built on OpenMPI+UCX, no libfabric

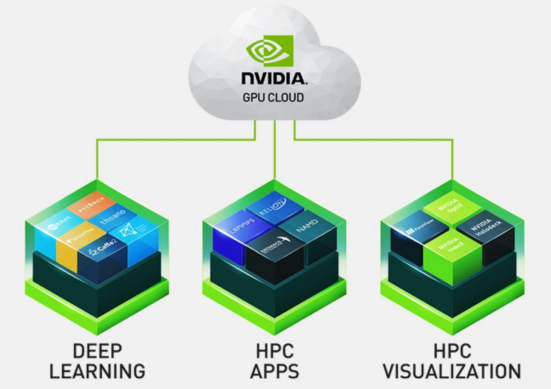
HPE Slingshot 11

- Latest generation high-performance interconnect for HPE systems
- Used in several world-class supercomputers

- Ethernet compatible, innovative congestion management, adaptive routing, QoS
- Uses libfabric for end-user API

NVIDIA GPU Cloud (NGC) images

- *"The NGC Catalog is a curated set of GPU-optimized software for AI, HPC and Visualization."*



- NGC container images are widely adopted to deploy ML/AI frameworks and workflows
- Built on OpenMPI+UCX, no libfabric
- NCCL implements connection-oriented transport APIs

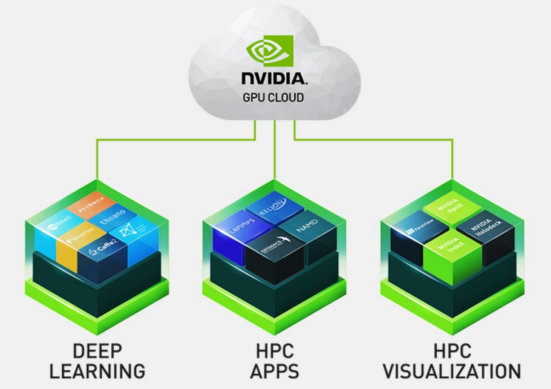
HPE Slingshot 11

- Latest generation high-performance interconnect for HPE systems
- Used in several world-class supercomputers

- Ethernet compatible, innovative congestion management, adaptive routing, QoS
- Uses libfabric for end-user API

NVIDIA GPU Cloud (NGC) images

- *"The NGC Catalog is a curated set of GPU-optimized software for AI, HPC and Visualization."*



- NGC container images are widely adopted to deploy ML/AI frameworks and workflows

- Built on OpenMPI+UCX, no libfabric

- NCCL implements connection-oriented transport APIs

HPE Slingshot 11

- Latest generation high-performance interconnect for HPE systems
- Used in several world-class supercomputers

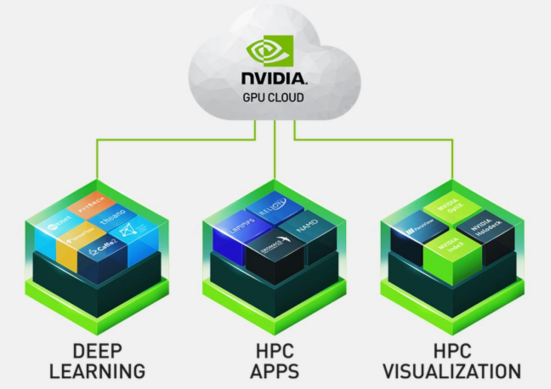
- Ethernet compatible, innovative congestion management, adaptive routing, QoS

- Uses libfabric for end-user API

- Libfabric provides connection-less interface

NVIDIA GPU Cloud (NGC) images

- *"The NGC Catalog is a curated set of GPU-optimized software for AI, HPC and Visualization."*



- NGC container images are widely adopted to deploy ML/AI frameworks and workflows

HPE Slingshot 11

- Latest generation high-performance interconnect for HPE systems
- Used in several world-class supercomputers

- Ethernet compatible, innovative congestion management, adaptive routing, QoS

- Built on OpenMPI+UCX, no libfabric ⟵?⟶ Uses libfabric for end-user API
- NCCL implements connection-oriented transport APIs ⟵?⟶ Libfabric provides connection-less interface

Connecting NCCL and libfabric

- NCCL supports external network plugins, which allow the library to leverage arbitrary or custom network technologies

- *AWS OFI NCCL is a plug-in which enables [...] to use libfabric as a network provider while running NVIDIA's NCCL based applications.* https://github.com/aws/aws-ofi-nccl

- Will it work with HPE's custom libfabric and proprietary CXI provider for Slingshot 11?
- Will it work in a container if mounted from the host?

3 key aspects

- **NGC images missing libfabric**: can a host libfabric work when completely mounted into a container without any previous installation?

- **Use the AWS OFI NCCL plugin**: can it work when bind mounted from the host into an environment where it was not compiled?

- **Control NCCL plugin selection:** NGC images ship with a default NCCL plugin. Can we drive NCCL to select the OFI plugin?

3 key aspects

- **NGC images missing libfabric**: can a host libfabric work when completely mounted into a container without any previous installation? **Yes!**
  - Libfabric must be visible to the container's dynamic linker (either through `LD_LIBRARY_PATH` or ldcache)

- **Use the AWS OFI NCCL plugin**: can it work when bind mounted from the host into an environment where it was not compiled?

- **Control NCCL plugin selection:** NGC images ship with a default NCCL plugin. Can we drive NCCL to select the OFI plugin?

3 key aspects

- **NGC images missing libfabric**: can a host libfabric work when completely mounted into a container without any previous installation? **Yes!**
  - Libfabric must be visible to the container's dynamic linker (either through `LD_LIBRARY_PATH` or ldcache)

- **Use the AWS OFI NCCL plugin**: can it work when bind mounted from the host into an environment where it was not compiled? **Yes!**
  - Plugin path in container must be in `LD_LIBRARY_PATH` to be found by NCCL

- **Control NCCL plugin selection:** NGC images ship with a default NCCL plugin. Can we drive NCCL to select the OFI plugin?

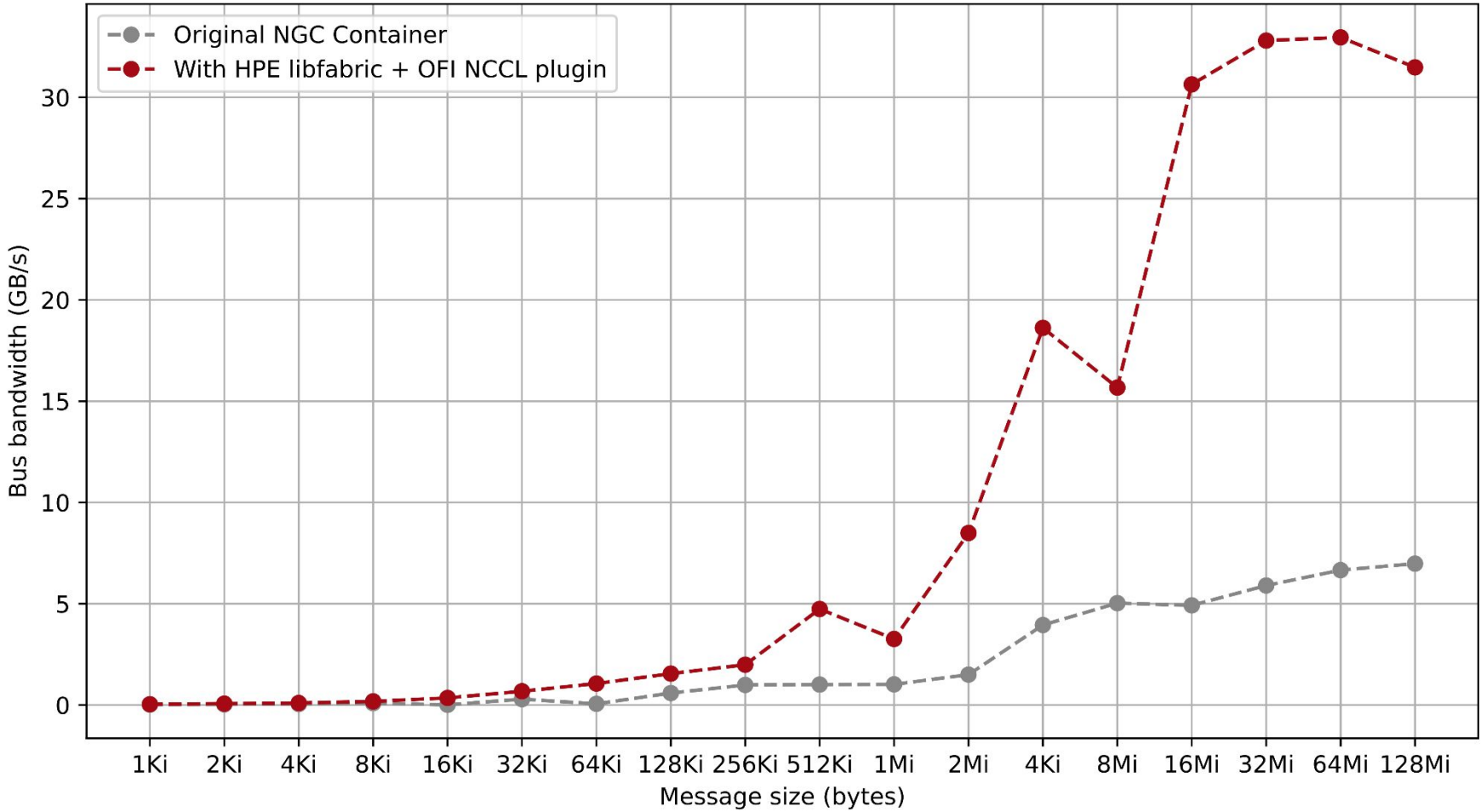
3 key aspects

- **NGC images missing libfabric**: can a host libfabric work when completely mounted into a container without any previous installation? **Yes!**
  - Libfabric must be visible to the container's dynamic linker (either through `LD_LIBRARY_PATH` or ldcache)

- **Use the AWS OFI NCCL plugin**: can it work when bind mounted from the host into an environment where it was not compiled? **Yes!**
  - Plugin path in container must be in `LD_LIBRARY_PATH` to be found by NCCL

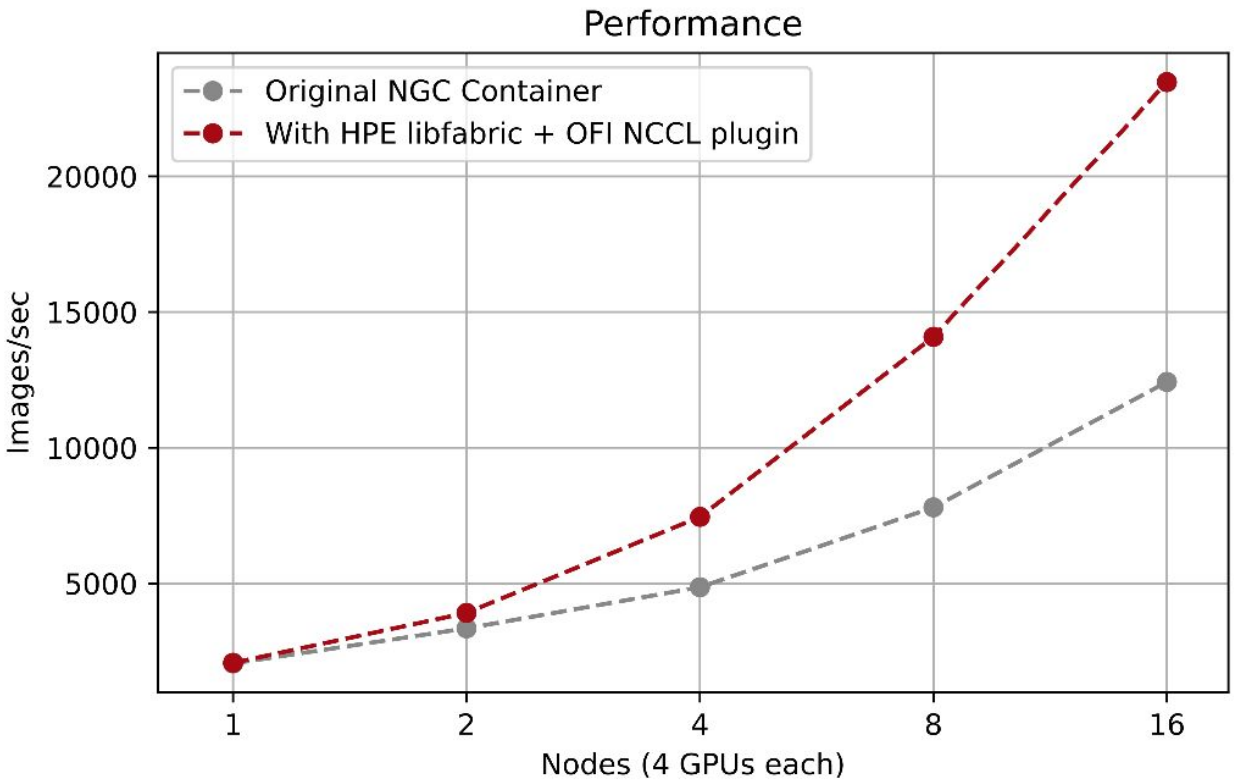
- **Control NCCL plugin selection:** NGC images ship with a default NCCL plugin. Can we drive NCCL to select the OFI plugin? **Yes!**
  - Mount plugin with an arbitrary suffix, e.g. `libnccl-net-ofi.so`
  - Set `NCCL_NET_PLUGIN` env var accordingly, e.g. `NCCL_NET_PLUGIN=ofi`

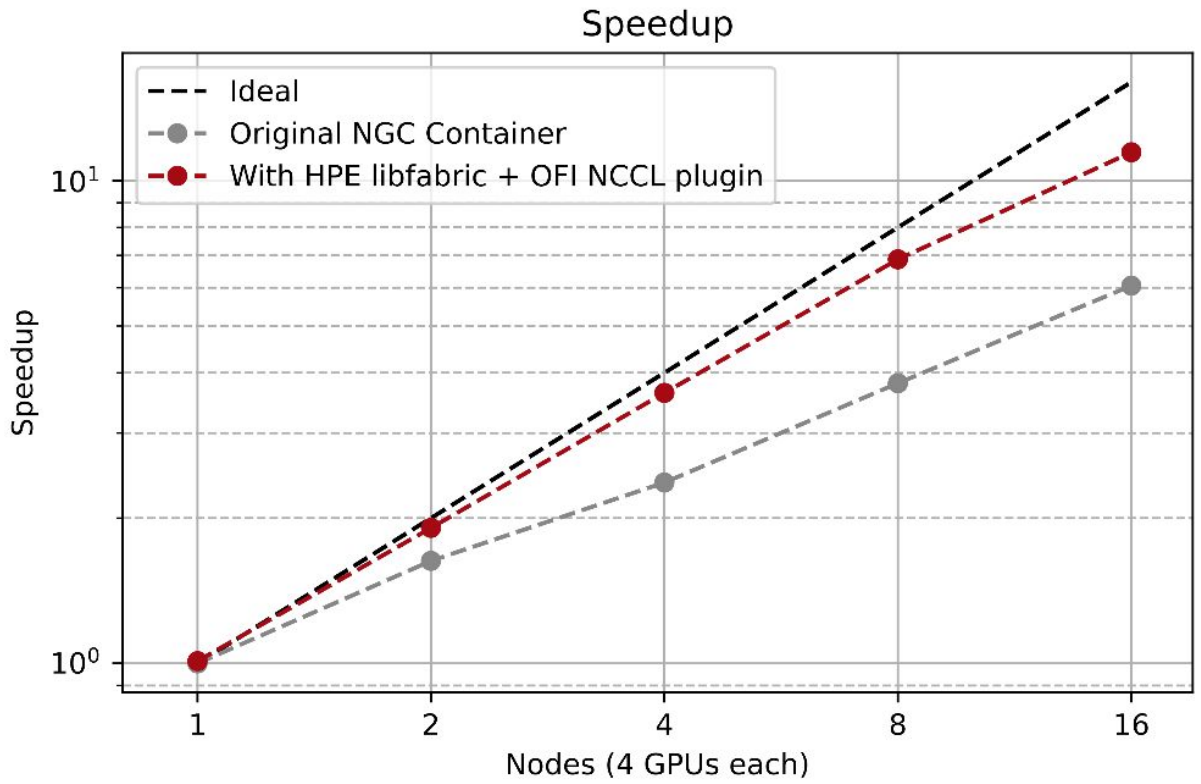
NCCL Tests: AllReduce bandwidth



Software: NGC PyTorch 22.12 base image, NCCL Tests 2.13.6, HPE libfabric 1.15.2.0
Test case: AllReduce performance test (8 physical nodes, 32 GPUs)
System: Alps Infrastructure - ML vCluster (4 x NVIDIA A100, HPE Slingshot 11 Interconnect)

PyTorch + Horovod: CNN training with synthetic data





Software: NGC PyTorch 22.12 base image, Horovod 0.28.1, HPE libfabric 1.15.2.0
Test case: Horovod PyTorch synthetic benchmark, ResNet-101
System: Alps Infrastructure - ML vCluster (4 x NVIDIA A100, HPE Slingshot 11 Interconnect)

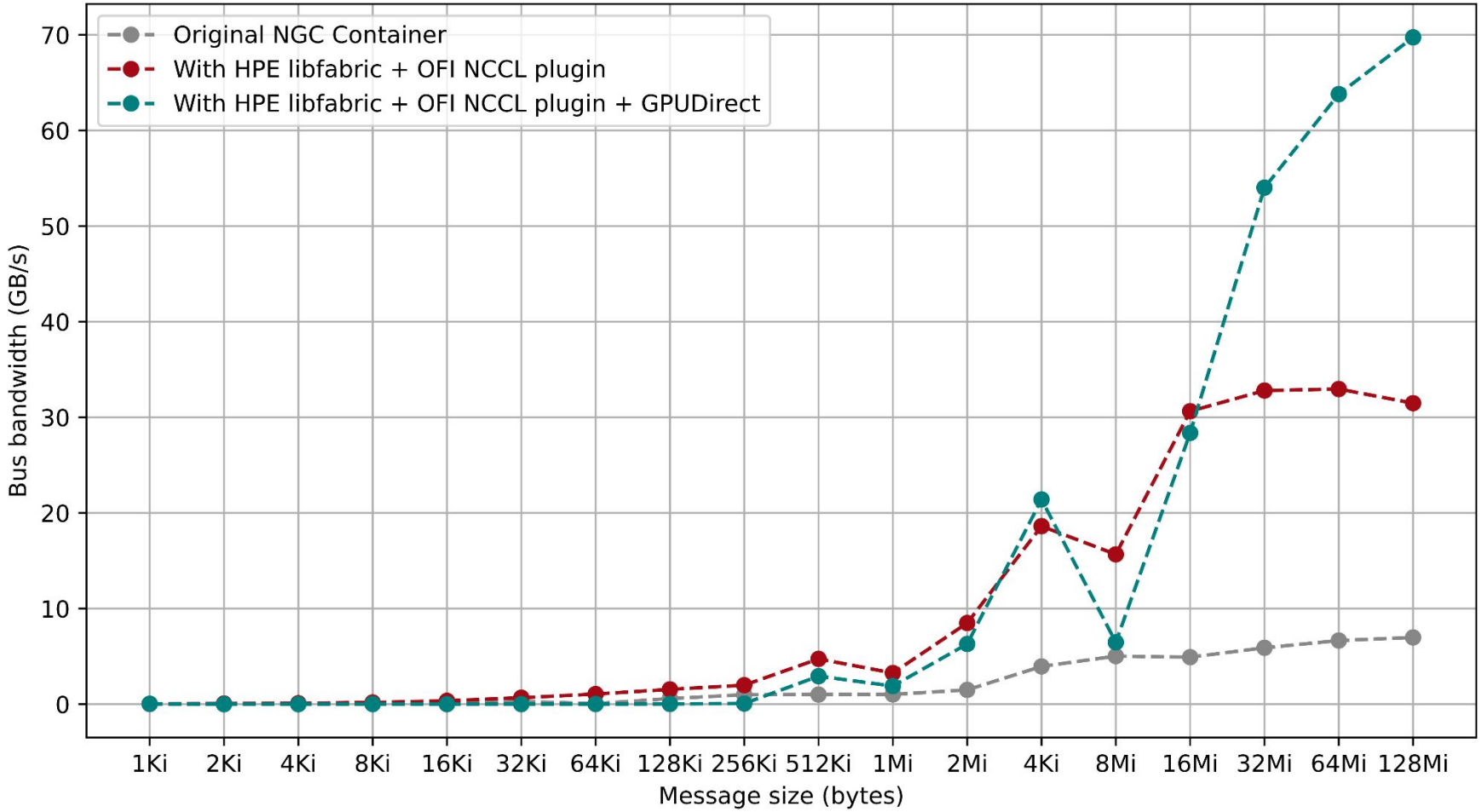
Enabling GPUDirect RDMA

- AWS OFI NCCL plugin
  - Use the `v1.6.0-hcopy` branch
  - Build with the `--with-gdrcopy` configure flag

- Ensure GDRCopy components are available in the container:
  - Driver kernel module: `/dev/gdrdrv`
  - User space library: `libgdrapi.so` (included in NGC image)

- Set environment variables for NCCL, libfabric and Slingshot:
  - ```
    CXI_FORK_SAFE="1"
    NCCL_NET_GDR_LEVEL="PHB"
    FI_CXI_DISABLE_CQ_HUGETLB="1"
    FI_MR_CACHE_MONITOR="userfaultfd"
    CXI_FORK_SAFE_HP="1"
    NCCL_CROSS_NIC="1"
    FI_CXI_DISABLE_HOST_REGISTER="1"
    ```
    ```

NCCL Tests: AllReduce bandwidth



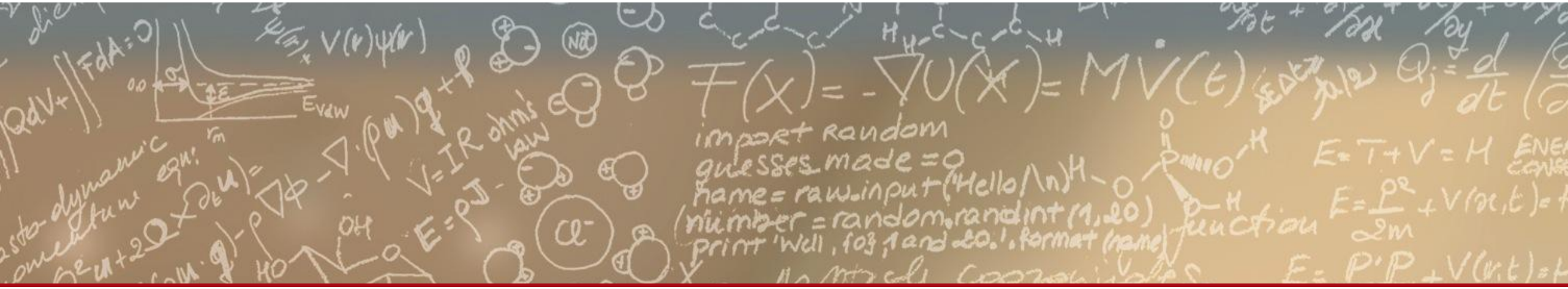
Software: NGC PyTorch 22.12 base image, NCCL Tests 2.13.6, HPE libfabric 1.15.2.0
Test case: AllReduce performance test (8 physical nodes, 32 GPUs)
System: Alps Infrastructure - ML vCluster (4 x NVIDIA A100, HPE Slingshot 11 Interconnect)

Takeaway messages

- It is possible to leverage Slingshot 11 for increased performance with NGC images, mounting host components from scratch:
  - libfabric to connect with Slingshot
  - NCCL plugin to interface with libfabric

- Benefit increases with scale.
  Harder to notice on multi-GPU nodes: NCCL uses shared memory intra-node

- **These are early results!**
  Vendors are actively working on improving support for this use case.
  Refer to HPE and NVIDIA for the latest recommended software releases and settings!



Thank you for your attention.