Skyway My Way! InfiniBand to Ethernet Optimization and Benchmarking

Joe Downs
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
joe@downs.dev

Brian Hsu
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
brianhsu1212@gmail.com

Jeremy Mao
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
jjmao616@gmail.com

Doug Egan
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
wegan@lanl.gov

Jesse Martinez
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
jmartinez@lanl.gov

***Abstract*—Modern high performance computing and AI/ML workloads increasingly require efficient bridges between InfiniBand (IB) compute fabrics and Ethernet storage systems to enable large data transfer capabilities. The NVIDIA Skyway is a gateway device designed to translate Internet Protocol over Infiniband (IPoIB) for external Ethernet services. This work seeks to evaluate and optimize the Skyway by benchmarking bandwidth performance, validating multi-tenant partition security boundaries, and identifying optimal deployment strategies for HPC workloads. Testing is performed on a nine node cluster with an InfiniBand, Ethernet, and Skyway switch. To test congestion, a modified version of GPCNeT is used to measure cluster bandwidth under traffic. NFS performance is benchmarked with various file transfers within both the Ethernet network and across the Skyway. Finally, the network is partitioned to separate traffic of two controller nodes, emulating a secure user compute system. Our experimentation paves the way for Skyway implementation on production clusters, and provides insight on hashing and storage I/O improvements for future work.**

***Index Terms*—High-performance computing, InfiniBand, Ethernet, IPoIB, NVIDIA Skyway, network gateway, NFS**

I. Introduction

High-performance computing (HPC) environments often combine low-latency InfiniBand (IB) compute fabrics with Ethernet-based storage and service networks. Connecting the two commonly requires dual-homed I/O servers, which increases hardware, cabling, and operational complexity. NVIDIA Skyway instead provides a layer-3 gateway that carries IPoIB traffic into Ethernet [1].

II. Experimental Platform

A. Testbed Topology

The experimental platform is shown in Fig. 1, consisting of two controller nodes, four compute nodes, two storage nodes, and one desktop endpoint, each connected to their respective switches with 200 Gb/s bandwidth cabling.

The NVIDIA Skyway operated as a Layer 3 gateway between the InfiniBand compute fabric and the Ethernet storage network, forwarding IPoIB traffic between the two environments.

This document was approved for unlimited release and given LA-UR-26-26452.

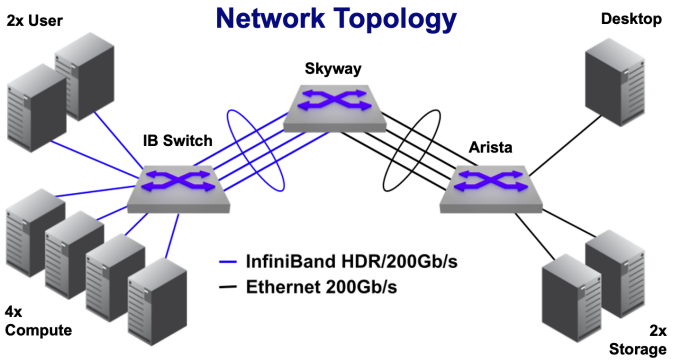


Fig. 1. Nine-node testbed connecting an HDR InfiniBand compute fabric to an Ethernet storage network through NVIDIA Skyway.

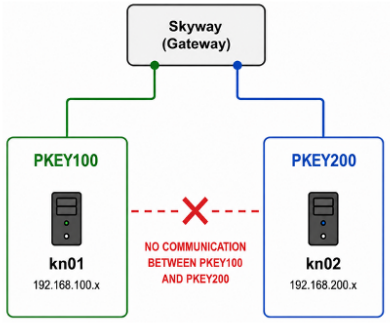


Fig. 2. Partition-key isolation remained intact across the Skyway gateway.

B. Partition Keys and Network Security

Three logical network domains were configured with InfiniBand partition keys (PKEYS) to separate the two workflow controllers. Each partition was assigned a distinct IPoIB interface and IP subnet, with static routes directing traffic from

each partition through the Skyway gateway to the Ethernet storage network[1]. Fig. 2 shows a high-level topology of the partitions.

III. Network Optimizations

Kernel and operating-system tuning set the CPU governor to performance mode, increased socket, TCP, and kernel ring buffers from 512 to 4096 bytes, and enabled jumbo frames on bonded interfaces. Aligning the InfiniBand and Ethernet maximum transmission units from 2044 to 4090 bytes reduced packet overhead. These changes raised single-source throughput from 123 to 170 Gb/s (iPerf) and multi-source throughput from 144 to 188 Gb/s, or approximately 38% and 30%, respectively. Both approached the 197 Gb/s direct-fabric baseline shown in Fig. 3. A remaining limitation was source-IP-only virtual-function hashing, which capped a single source near 170 Gb/s because layer-3/layer-4 flow hashing was unavailable for distribution.

IV. Congestion Tests

A six-node modified GPCNet test used two measurement nodes (one storage, one compute) and four traffic-generating nodes (one storage, three compute) [4]. Concurrently, the traffic generators ran GPCNeT's all-to-all, two-sided incast, RMA Put incast, and RMA Get broadcast patterns, creating mixed many-to-many, many-to-one, and one-to-many MPI traffic across the compute–gateway–storage path. Across 6 to 384 MPI ranks, isolated and congested bandwidth remained nearly identical, indicating that the simulated competing traffic had negligible effect on the measured path (Fig. 4).

V. Network File System Performance

NFS performance was evaluated using fio random reads with one job and an I/O depth of four [5], [6]. A direct Ethernet path was compared with an InfiniBand-to-Ethernet path routed through the Skyway.

A. File-Size Sensitivity

A file-size test fixed the block size at 4 KiB and varied the file size from 4 KiB to 2 GiB. As shown in Fig. 6, bandwidth increased with file size before stabilizing near 1.5–2.0 Gb/s. Both paths followed similar trends, indicating that small-request filesystem and protocol overheads, rather than Skyway forwarding, were the primary limitation.

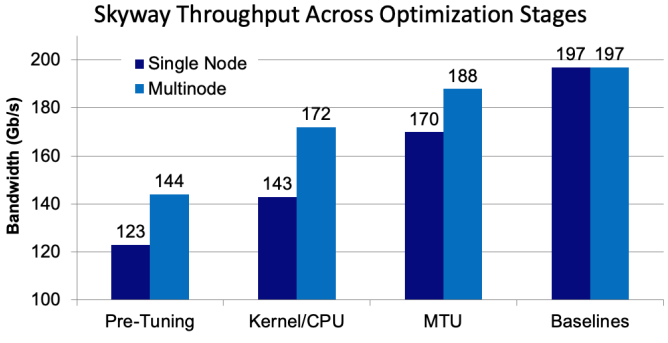


Fig. 3. iPerf bandwidth test results. Incremental IPoIB tuning compared with the direct HDR InfiniBand baseline.

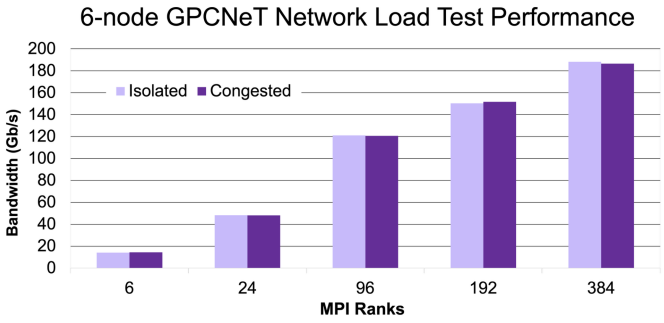


Fig. 4. Modified GPCNet load test under isolated and congested conditions.

B. Block-Size Sensitivity

A block-size sweep fixed the file size at 2 GiB and varied the block size from 1 KiB to 16 MiB. Fig. 5 shows that throughput increased substantially with larger requests, peaking at a 4 MiB block size. The direct Ethernet path reached approximately 114 Gb/s, while the Skyway path reached approximately 93 Gb/s. These results indicate that storage access is more sensitive to I/O configuration than to the gateway itself, still supporting high-bandwidth storage transfers.

VI. Results

Multi-source throughput increased to 188 Gb/s, within approximately 5% of the 197 Gb/s native HDR baseline shown in Fig. 3. The remaining single-source limitation was associated with source-IP-only virtual-function hashing rather than the gateway's aggregate forwarding capacity.

The congestion tests showed that throughput remained stable under the tested background load, while the partitions preserved separation between the two InfiniBand subnets across the shared gateway infrastructure. The NFS measurements showed small requests were limited primarily by protocol overhead, while larger blocks still reached high throughput.

VII. Conclusion

NVIDIA Skyway provided an effective interoperability path between an HDR InfiniBand compute fabric and an Ethernet storage network. After host and MTU tuning, throughput

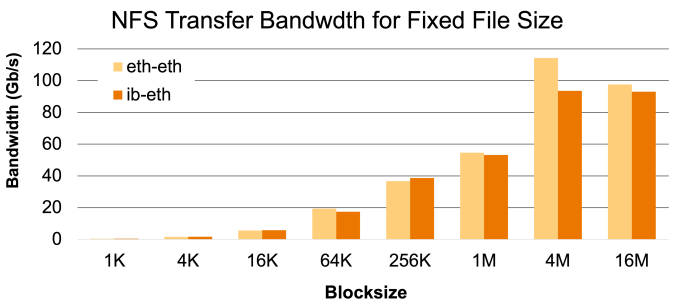


Fig. 5. NFS random-read bandwidth versus block size using a 2-GiB file.

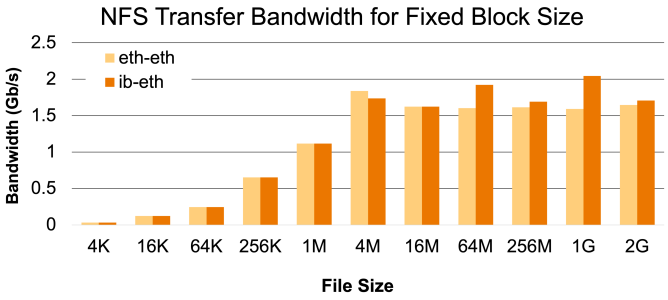


Fig. 6. NFS random-read bandwidth versus file size.

approached the native fabric rate, remained stable under the tested congestion, and preserved isolation between independent InfiniBand partitions.

The primary remaining constraints were single-flow hashing and storage-I/O efficiency rather than the gateway's overall forwarding capacity. These results support Skyway as a practical alternative to deploying dual-homed I/O servers in mixed-fabric HPC environments.

REFERENCES

[1] NVIDIA Corporation, *NVIDIA Skyway InfiniBand-to-Ethernet Gateway User Manual*, Mar. 3, 2025. Accessed: Jul. 20, 2026. [Online]. Available: https://networking-docs.nvidia.com/skywayhw/

[2] V. Kashyap, "IP over InfiniBand (IPoIB) architecture," RFC 4392, Apr. 2006, doi: 10.17487/RFC4392.

[3] Iperf2 Project, *Iperf 2*, ver. 2.1.6, Dec. 2021. Accessed: Jul. 20, 2026. [Online]. Available: https://sourceforge.net/projects/iperf2/files/iperf-2.1.6.tar.gz/download

[4] S. Chunduri et al., "GPCNeT: Designing a benchmark suite for inducing and measuring contention in HPC networks," in *Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2019, Art. no. 42, pp. 1–33, doi: 10.1145/3295500.3356215.

[5] T. Haynes and D. Noveck, Eds., "Network File System (NFS) version 4 protocol," RFC 7530, Mar. 2015, doi: 10.17487/RFC7530.

[6] J. Axboe, *Flexible I/O Tester (fio)*, ver. 3.35, May 2023. Accessed: Jul. 20, 2026. [Online]. Available: https://github.com/axboe/fio/releases/tag/fio-3.35