Benchmarking Ethernet Interconnect for HPC/AI workloads

Lorenzo Pichetti* University of Trento
Daniele De Sensi* Sapienza University of Rome
Karthee Sivalingam Open Edge and HPC Initiative, Huawei
Stepan Nassyr ParTec AG, FZ Jülich
Daniele Cesarini CINECA
Matteo Turisini CINECA
Dirk Pleiter Open Edge and HPC Initiative, KTH
Aldo Artigiani Huawei Datacom
Flavio Vella* University of Trento

***Abstract*—Interconnects have always played a cornerstone role in HPC. Since the inception of the Top500 ranking, interconnect statistics have been predominantly dominated by two competing technologies: InfiniBand and Ethernet. However, even if Ethernet is very popular due to versatility and cost-effectiveness, InfiniBand used to provide higher bandwidth and continues to feature lower latency. Industry seeks for a further evolution of the Ethernet standards to enable fast and low-latency interconnect for emerging AI workloads by offering competitive, open-standard solutions. This paper analyzes the early results obtained from two systems relying on an HPC Ethernet interconnect, one relying on 100G and the other on 200G Ethernet. Preliminary findings indicate that the Ethernet-based networks exhibit competitive performance, closely aligning with InfiniBand, especially for large message exchanges.**

I. Introduction

As artificial intelligence (AI) and high-performance computing (HPC) continue to push the boundaries of computational capabilities, the demands on networking infrastructure have become increasingly stringent. Modern AI workloads, including large language models and deep learning recommendation models, require significant computational resources distributed across large clusters of GPUs. The efficiency of these workloads critically depends on the underlying network's ability to move data with minimal latency and maximal bandwidth [1], [2], [3].

As we move towards the post-exascale era, the scale of the clusters is growing and has resulted in higher requirements on network performance, scalability, and resilience.

The development of *Remote Direct Memory Access* (RDMA) architectures [4] was, in particular, driven by the goal of reducing latency and increasing bandwidth. With RDMA, part of the data transport is offloaded to the network interface card (NIC) [5], allowing servers to read and write memory locations on other servers without (almost) any host involvement, thus reducing latency and CPU utilization. This enables communications between servers within 1 µs [6]. Traditionally, RDMA was often implemented through proprietary protocols [7], [8], [9]. However, nowadays, there is an increasing effort towards standardization.

For example, the InfiniBand (IB) [10] specification is maintained and developed by the InfiniBand Trade Association (IBTA) [11]. Nevertheless, InfiniBand hardware is only provided by a few vendors, increasing the risk of vendor lock-in. For this reason, the industry is moving towards RDMA solutions relying on Ethernet, due to the wider availability of Ethernet hardware, its standardization, and the fact that it can enable convergence with other types of traffic (e.g., north-south traffic in data centers) [12]. One step towards this direction is represented by *RDMA over Converged Ethernet* (RoCE) (also maintained by IBTA) [13]. RoCE is a specification of InfiniBand running on top of Ethernet rather than the IB link layer. Last, efforts such as iWARP [14], [15] run an RDMA protocol on top of TCP but suffer, however, from TCP limitations [16].

One challenge for Ethernet-based RDMA is packet loss, which can severely impact performance. Despite the efforts, available solutions are still far from being optimal. Namely, RoCE works better on lossless networks. However, lossless Ethernet heavily relies on *Priority Flow Control* (PFC), which has been shown to require excessive buffering, and creating congestion trees and PFC storms and deadlocks [12], [17]. Although some of these issues are partially addressed by congestion control algorithms such as *Data Center Quantized Congestion Notification* (DCQCN) [18], the co-existence with other types of traffic is often problematic and can create unfairness [19].

Thus, although Ethernet-based network solutions have a good chance of becoming competitive to HPC-optimised network solutions like InfiniBand, we are not there, yet [20], [21]. For these reasons, some technologies like Slingshot provide a custom *high-performance* Ethernet implementation, optimized but fully interoperable with Ethernet [22]. On the other hand, initiatives such as the UltraEthernet Consortium (UEC) [23] aim to leverage the widely used Ethernet technologies and its standards to define a new Ethernet tailored for HPC and AI applications.

This work is motivated by the growing convergence of HPC and AI networking requirements and the need for more scalable, cost-effective, and high-performance network solutions.

*Corresponding authors.

979-8-3503-5554-3/24/$31.00 ©2024 IEEE
DOI 10.1109/SCW63240.2024.00124

	Nanjing	Haicgu (Ethernet partition)	Haicgu (InfiniBand partition)
CPU	2× 48-core Kunpeng920 2600MHz	2x64-core HiSilicon *Kunpeng 920-6426*	
NICs	Mellanox 200G(MCX653106A-HDAT)	Mellanox/NVIDIA ConnectX-5 MCX555A-ECAT (4x25)100GE/100GBit EDR IB	
Interconnect	200GE connect with NIC	100GE connect with NIC	100Gbps InfiniBand connect with NIC
Topology	two-level fat-tree with CE9855 switches	single-level leaf-switch CE8850	single-level leaf-switch InfiniBand EDR
Software Environment	OpenMPI 4.0.3 + GCC 9.3.0 (UCX 1.15.0 , without verbs, MCA pmix v2.1.0)	GCC 14.1.0 + OpenMPI 5.0.3 (relying on UCX 1.16.0, UCC 1.3.0, PMIx 5.0.2)	

TABLE I: Main characteristics of the analyzed systems.

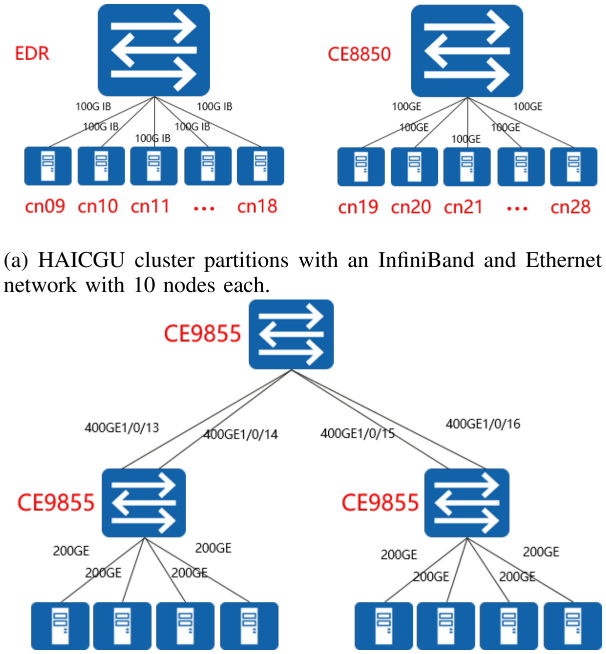
The main contribution of this work is the evaluation of new HPC-tailored Ethernet technologies; in particular, we focus on analyzing Huawei's HPC Ethernet [24]. We compare it with InfiniBand in a series of MPI-based benchmarks, to determine whether advanced Ethernet technology can meet the demands of next-generation AI and HPC workloads. Both clusters provide a controlled environment for directly comparing different interconnect technologies.

The rest of this paper is organized as follows. In Section II, we describe the characteristics and configurations of the two HPC systems used in our benchmarks. Section III outlines the benchmarking methodology and the performance tuning applied to optimize the testing environment. It also presents the experimental results, providing a comparative analysis of Ethernet and InfiniBand performance across various scenarios. Finally, in Section IV, we discuss the paper's findings and their implications for future HPC and AI networking solutions.

II. Systems Description

In the following, we describe the main features of Huawei's HPC lossless Ethernet fabric [24], and the analyzed systems' main characteristics (summarized in Table I). Huawei's HPC Ethernet runs on an open architecture and ecosystem in which the involved technologies, devices, and components are all based on Ethernet standards. On top of that, the following advanced proprietary features (not part of the Ethernet specification) are provided, to support the needs of HPC and AI workload requirements:

- **Loss prevention**: By exploiting *Priority-based Flow Control* (PFC), packet losses are avoided, thus providing the lossless, low-latency, and high-throughput network environment needed for RoCEv2 traffic, meeting high-performance requirements.
- **PFC deadlock prevention**: Service flows that may cause deadlocks are identified, and queue priorities of these flows are changed to prevent PFC deadlocks.
- **Artificial Intelligence Explicit Congestion Notification (AI ECN)**: AI ECN can intelligently adjust the ECN thresholds of lossless queues based on the observed traffic characteristics to ensure low latency and high throughput with zero packet loss, maximizing the performance of lossless services.
- **Network Scale Load Balancing (NSLB)**: To adaptively route the traffic over the network based on the observed traffic characteristics.
- **ECN Overlay**: applies ECN to a VXLAN network, enabling the traffic receiver to detect congestion on the overlay network in a timely manner and instruct the traffic sender to reduce its packet sending rate to relieve network congestion.



(a) HAICGU cluster partitions with an InfiniBand and Ethernet network with 10 nodes each.

(b) Nanjing lab setup showing CE9855 switches connected in a spine-leaf configuration to 8 nodes.

Fig. 1: Interconnect architectures of the clusters used for benchmarking.

A. OEHI (Ethernet/InfiniBand)

The Huawei AI and Computing at Goethe University (*HAICGU*) [25] is installed at the Goethe University of Frankfurt and maintained by the Open Edge and HPC Initiative (OEHI). This cluster is designed with two partitions: one for InfiniBand (*cn-ib*) and another for Ethernet (*cn-eth*) to support their respective interconnects. Each partition consists of 10 nodes.

Node architecture: Each compute node features two sockets equipped with 2× Kunpeng 920-6426 CPUs (64-bit ARMv8.2-A cores running at 2.6 GHz). Each node has 16× 8 GiB DIMMs of registered ECC DDR4-2933 memory, for a total of 128 GiB CPU memory. There are 40× PCIe v4.0 lanes, used for communications between the host CPU and the Mellanox ConnectX-5 NIC.

Inter-node connectivity: All nodes are connected via a Huawei Switch S5735-S48T4X 10 GbE Ethernet switch for deployment and management. The *cn-ib* and *cn-eth* partitions each consist of 10 compute nodes with identical hardware specifications and software stacks. In the *cn-ib* partition, the 10 nodes are connected via a non-blocking Mellanox Switch-IB 2 EDR (MSB7890-ES2F) 100 Gb/s fabric, enabling high-bandwidth, low-latency communication. In the *cn-eth* partition, nodes are interconnected through a 100GE network using a Huawei CE8850 (CE8850-64CQ-E) switch. The Mellanox ConnectX-5 adapters support both IB and Ethernet traffic and have been configured accordingly for each partition.

Switch properties: CloudEngine 8850 [26] are Ethernet switches designed for data centers, featuring high-performance, high-density, and low-latency. Built on an advanced hardware structure, the switches support high-density 100GE/40GE/25GE/10GE ports. The CloudEngine 8850-64CQ-EI supports 12.8 Tbps switching capacity, 4482 Mpps forwarding performance, and L2/L3 line-speed forwarding. The CloudEngine 8850-64CQ-EI supports up to 64× 100GE QSFP28 ports and 64× 40GE QSFP+ ports and can function as a core or aggregation switch.

B. Nanjing

The HPC environment in Nanjing uses the spine-leaf architecture and consists of three Huawei CE9855 switches and 8 compute nodes. One CE9855 functions as the spine node, and two CE9855s function as leaf nodes. Each leaf node is connected to four compute nodes, and the network is wired as a non-blocking fat tree.

Node architecture: Each server has 2× sockets with one 48-core CPU per socket. Each node has 16× 32 GiB CPU memory organized in 16 slots.

Inter-node connectivity (Ethernet): Nodes are interconnected through a 200GE network, and each node is equipped with one Mellanox ConnectX-6 NIC. The spine-leaf architecture is used, using 400GE connectivity between spine and leaf switches.

Switch properties: CloudEngine 9855 [27] series switches are next-generation high-performance and high-density 400GE access switches designed for HPC and AI scenarios. They have an advanced hardware architecture, offer high-density 400GE access ports, and support 400GE uplink ports. They support a maximum of 32× 400GE high-performance QSFP-DD ports. Each 400GE QSFP-DD port is backward compatible with 200GE/100GE/40GE interfaces, and can be split into four 100GE ports or two 200GE ports. The split ports support IEEE 1588v2 (PTP) and provide flexibility in networking. A 400GE port working as a 200GE/100GE/40GE port cannot be split.

C. Performance Tuning

To maximize the performance of the ConnectX NICs, we transitioned from the default inbox drivers to OFED drivers. Table II summarizes all the applied tuning configurations.

Action	**Command**
Check whether the NIC is working on RoCE v2	`cma_roce_mode -d mlx5_0 -p 1`
Ensure that queue 3 enables PFC function	`mlnx_qos -i eth4 --pfc 0,0,0,1,0,0,0,0 --trust dscp`
Change maximum packet size to 9000 bytes	`ifconfig eth4 mtu 9000`

TABLE II: System tuning operations.

III. Experimental results

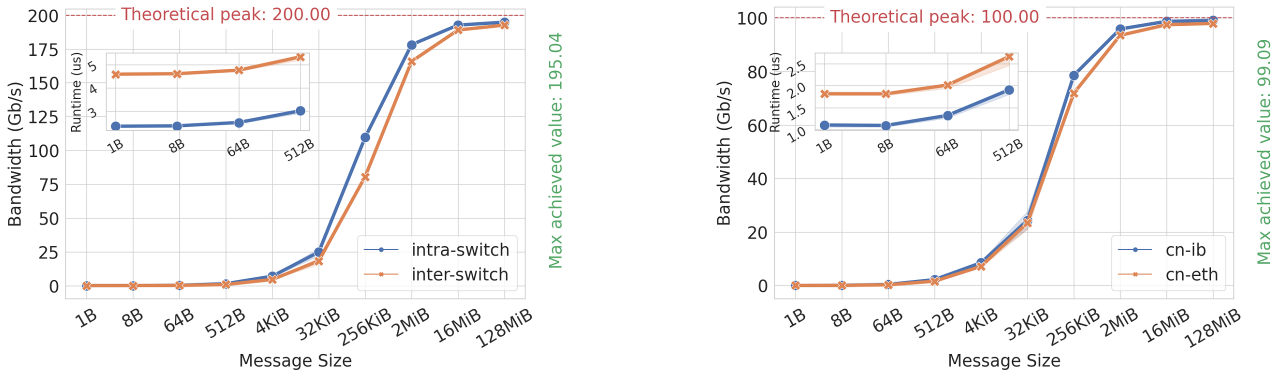
A. Benchmarking Methodology

All experiments were conducted with exclusive execution, ensuring that noise from concurrently running jobs did not affect results. We always use a single MPI process per node. All the experiments did not include the communicators' creation time and, depending on the buffer size, were repeated between 100 times and 1,000 times. Regarding collective communications, we report the maximum time (or minimum throughput) across all the involved ranks [28]. The bandwidth is always reported in Gbit s^{-1} and, regardless of the benchmark, we compute it as the aggregated message size transmitted on the wire, divided by the maximum time employed by all the processes to complete the operation. Unless specified otherwise, the theoretical peak is equivalent to the system's unidirectional data-transfer peak of the NIC (100 Gbit s^{-1} in case of the HAICGU and 200 Gbit s^{-1} in case of the Nanjing lab cluster).

B. Point-to-point Performance

The *point-to-point* tests aim to measure the latency and the maximum bandwidth that can be achieved between two nodes. During the *point-to-point* test, we assign two processes to two different nodes, and we make them exchange a fixed-size buffer several times. This test represents the simplest benchmarked scenario and, since the network is non-blocking, we expect to fully saturate the available bandwidth. Figure 2 presents the peer-to-peer results for the HAICGU and Nanjing clusters.

On the HAICGU cluster, we compare the Ethernet network performance with InfiniBand performance (respectively *cn-eth*, *cn-ib*); these data underline how the two interconnections expose qualitative and quantitative comparable performance over large messages. Whereas InfiniBand always outperforms Ethernet, on messages larger than 256 KiB, this difference in performance is always lower than 4%. Conversely, for small messages (smaller than 512 B), InfiniBand significantly outperforms Ethernet up to a factor of 1.6×. This is partially due to the higher overhead introduced by the larger headers (Ethernet, IP, and UPD) in the Ethernet/RoCEv2 case.



(a) Nanjing

(b) HAICGU

Fig. 2: *point-to-point* results over the considered systems. The x-axis represents the buffer size exchanged by the two MPI processes; the y-axis ranges from 0 to the theoretical peak and represents the achieved bandwidth.

Observation 1: Measured performance gap between Ethernet and InfiniBand on message sizes greater equal than $32\,\mathrm{KiB}$ is always lower than 4%.

Because in the Nanjing cluster the nodes can be connected either through one or three switches, we analyze how the network distance affects the *point-to-point* performance. Figure 2a report as *intra-switch* the *point-to-point* bandwidth related to nodes connected to the same switch and as *inter-switch* the one for nodes that communicate through the spine switch.

As expected, the impact of network distance is significant on small messages (~2× slow down for 1-512 B message sizes) and becomes smaller as the messages grow (~1.4× for 4-256 KiB and ~1.04× for 1-128 MiB).

The impact of traversing through multiple switches can be assessed by comparing the *intra-switch* and *inter-switch* results. Since all the switches and all the links are equivalent, by halving the difference between the *intra-switch* and the *intra-switch* time for transferring 1 B, we can estimate the latency for crossing one link and one switch as $1.11\,\mu\mathrm{s}$.

C. Incast micro-benchmark

In contrast to the *point-to-point* scenario, the *incast* micro-benchmark stresses network congestion control by directing all messages to a single process. It synthetically simulates a completely unbalanced communication scheme which often occurs in practice and it has been shown to be a challenging scenario for congestion control algorithms [29]. During that test, we first select a main process i among the n involved ones, and then all the other $n-1$ processes will send a fixed-size buffer to i. Since all the messages are received by the same process, the theoretical bottleneck is the injection bandwidth of any given process divided by $n-1$, since the receiver node bandwidth must be shared among all the sender processes.

Figure 3 shows the results for the incase test using three different setups: 4 nodes of the Nanjing lab cluster, the full Nanjing system, and the full HAICGU system. On Nanjing, in the 8 nodes case half of the traffic is forced to cross the spine switch. When using 4 nodes we consider both the case where the four processes are under the same lower-level switch, and then the case with two processes per leaf (*i.e.* a part of the communication is forced to pass through the spine switch). It is worth recalling that in the HAICGU cluster, all the nodes are under the same switch.

On HAICGU, differently from *point-to-point* tests, the *incast* comparison shows that Ethernet is competitive also for small message sizes; the Ethernet/InfiniBand gap is lower than 20% on all the considered sizes and negligible or absent ($\leq 1\%$) on messages greater than $32\,\mathrm{KiB}$.

Nanjing's results confirm the same behaviors of *point-to-point* tests; tests can achieve near-optimal performance (96% of theoretical peak), and the impact of network distance is significant for small messages and is highly amortized as the message size grows.

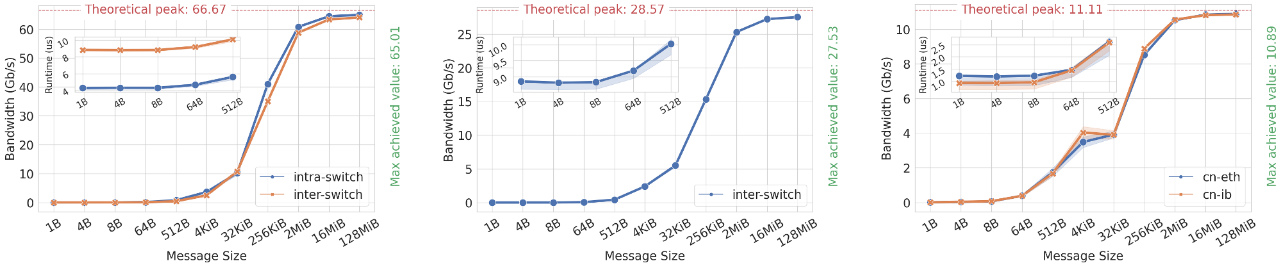
Overall, *incast* tests highlights that Ethernet does not struggle with unbalanced communication and traffic congestion, in fact, in all the instances, tests always achieve at least the 96% of the maximum theoretical bandwidth.

Observation 2: On incast traffic, Ethernet is comparable with InfiniBand. For all the message sizes, the performance gap is always lower than the 20%. Furthermore, as the message size grows over $32\,\mathrm{KiB}$, this gap becomes negligible ($\leq 1\%$).

D. All-reduce and all-to-all communication

All-reduce and *all-to-all* are widely used primitives in distributed programming, and they represent the main communication bottleneck in many real-world applications. Due to their wide applicability and efficiency, they are often among the most commonly used collectives across many application domains, including many HPC and AI applications [30].

During the *all-to-all* collective, all the involved processes exchange a fixed-size message with all the other involved



(a) Nanjing Incast 4 nodes

(b) Nanjing Incast 8 nodes

(c) HAICGU Incast 10 nodes

Fig. 3: *incast* results over the considered systems, Nanjing was measured on both the system-wide and the leaf-wide test cases. The x-axis represents the message size that each MPI process sends to the main process; the y-axis ranges from 0 to the theoretical peak and represents the achieved bandwidth. The theoretical peak is computed as the injection bandwidth of any given process divided by the number of sender processes.

processes; so, each of the n processes will concurrently send and receive $n-1$ messages, one per process. In contrast, in *all-reduce*, all the processes own a different local buffer and, once the collective operation is complete, all the processes must own the same buffer obtained as the reduction of all the starting buffers.

We measure performance using *goodput*, by assuming the bandwidth-optimal implementation of the collective is used. We compute it by dividing the assumed number of exchanged bytes by the achieved time-to-solution. Essentially, goodput is equivalent to bandwidth if and only if the most efficient collective implementation is used. For example, for allreduce we assume that the ring algorithm is used, and that $2b\cdot(n/(n-1))$ bytes are transmitted, where b is the size of the vector to be reduced and n is the number of ranks participating in the allreduce [31], [32]. For the *all-to-all*, we consider each process sends $b\cdot(n-1)$ bytes, where b is the number of elements received from any process.

Figure 4 reports the collectives' performance over the two considered systems.

1) All-to-all: Figure 4b shows the *all-to-all* performance over the Nanjing cluster. As for the previous tests, the achieved bandwidth grows until reaching the 66% of the theoretical peak for message sizes of 16 MiB.

HAICGU cluster exposes similar behaviours but it is interesting to note that for messages of size 128 MiB, the Ethernet regression to 58% of peak is way more drastic than the 81% achieved by InfiniBand; this is the bigger gap that our analysis found between Ethernet and InfiniBand performances, and we are currently running further analysis to better understand this issue.

2) All-reduce: On *all-reduce*, we encountered an unexpected upper bound of approximately 20 Gbit/s on both the Nanjing lab cluster and HAICGU. While network congestion and reduction operations can introduce some overhead, they can not justify the substantial gap observed between *all-to-all* and *all-reduce*. While switch-level counters did not indicate any bottlenecks at the hardware level, the error appears to be systematic and related to both InfiniBand and Ethernet, strongly suggesting that the issue is independent of the network and likely related to some host-specific problems. To further validate our hypothesis, we connected two hosts directly and still observed the same upper bound limit on goodput, around 20 Gbit/s. This hypothesis is further supported by the fact that the observed bottleneck is consistent across both the Nanjing lab cluster and HAICGU, which have identical hosts but completely different networks. Additionally, the bottleneck remains unaffected by the number of involved processes. We are continuing to investigate the issue and are currently conducting additional analysis.

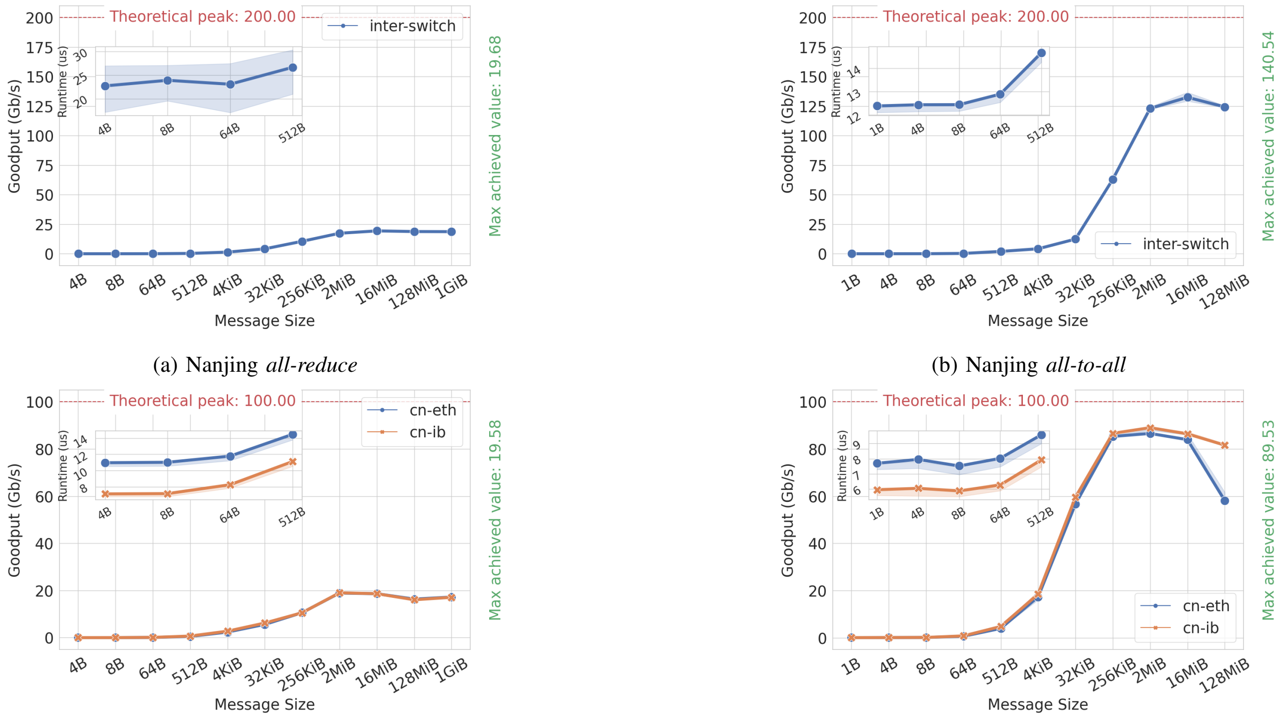
Observation 3: With the exception of *all-to-all* with message size 128 MiB, for *all-to-all* and *all-reduce* communication patterns with message sizes larger than 32 KiB, the difference between Ethernet and InfiniBand performance is below 3%.

IV. Conclusions

In this work, we investigate Ethernet technology as an alternative to the InfiniBand interconnects for HPC and AI workloads.

We perform our analysis on two systems: the HAICGU and the Nanjing clusters. On the one hand, HAICGU provides two equivalent partitions to benchmark Ethernet and standard InfiniBand interconnections up to 10 computation nodes, on the other, the Nanjing lab's cluster comprises 8 computation nodes interconnected by an Ethernet-based network with a two-level fat-tree topology.

Over these systems, we tested *point-to-point*, *incast*, *all-to-all*, and *all-reduce*, which are the most important and commonly used communication patterns in HPC and AI workloads. Although the poor performance measured for *all-reduce* requires further investigation, our results demonstrate that on messages larger then 32 KiB, and across all the performed tests, Ethernet and InfiniBand exhibit a performance gap of less than 10%. However, from a latency prospective, InfiniBand still significantly outperforms Ethernet by a factor of



(a) Nanjing *all-reduce*

(b) Nanjing *all-to-all*

(c) HAICGU *all-reduce*

(d) HAICGU *all-to-all*

Fig. 4: Collective results over the system-wide test cases. The x-axis represents the message size that each MPI process exchanges with each other process; the y-axis ranges from 0 to the theoretical peak and represents the achieved bandwidth. The significant gap between the expected and measured *all-reduce* performance is under investigation and related to host configuration issues rather than the network.

approximately $1.4\times$. These results suggest that Ethernet shows significant promise and performs comparably to InfiniBand in many scenarios, but it still requires further development and optimization to fully meet the demands of high-performance computing and AI workloads, especially in preparation for the upcoming Ultra Ethernet specifications.

This work opens up multiple directions for further exploration. We aim to extend our findings by applying them to real-world HPC and AI applications, thereby demonstrating the relevance of our micro-benchmarks. Additionally, in response to the growing interest in GPU-centric communication [33], [34] within HPC and AI workloads, we plan to test the investigated interconnections in a GPU-centric scenario. This will help us assess how our results vary with different system configurations and setups.

Acknowledgement

We thank OEHI and Huawei Technologies DataCom for supporting this study by providing access to the HAICGU and Nanjing lab clusters. In particular, we extend our gratitude to the Huawei Technologies DataCom PoC team—Wenjia Lai, Dancheng Zhang, Xiaolong Zheng, and Yang Pan—for their fundamental support in operating the Nanjing lab clusters. Furthermore, we gratefully acknowledge the Goethe University of Frankfurt for hosting the HAICGU. This work has been partially funded by Sapienza University under the SEED-2022 and ”Progetti Grandi 2023” funding schemes. Daniele De Sensi and Flavio Vella are members of *Gruppo Nazionale Calcolo Scientifico - Istituto Nazionale di Alta Matematica* (GNCS-INdAM). Flavio Vella has been partially supported by ICSC (National Research Center in HPC, Big Data and Quantum Computing), funded by the European Union – NextGenerationEU action.

References

[1] Q. Hu, P. Sun, S. Yan, Y. Wen, and T. Zhang, “Characterization and prediction of deep learning workloads in large-scale gpu datacenters,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC ’21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3458817.3476223

[2] M. Jeon, S. Venkataraman, A. Phanishayee, u. Qian, W. Xiao, and F. Yang, “Analysis of large-scale multi-tenant gpu clusters for dnn training workloads,” in *Proceedings of the 2019 USENIX Conference on Usenix Annual Technical Conference*, ser. USENIX ATC ’19. USA: USENIX Association, 2019, p. 947–960.

[3] Q. Weng, W. Xiao, Y. Yu, W. Wang, C. Wang, J. He, Y. Li, L. Zhang, W. Lin, and Y. Ding, “MLaaS in the wild: Workload analysis and scheduling in Large-Scale heterogeneous GPU clusters,” in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. Renton, WA: USENIX Association,

Apr. 2022, pp. 945–960. [Online]. Available: https://www.usenix.org/conference/nsdi22/presentation/weng

[4] A. Kalia, M. Kaminsky, and D. G. Andersen, "Design guidelines for high performance {RDMA} systems," in *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, 2016, pp. 437–450.

[5] Z. Wang, L. Luo, Q. Ning, C. Zeng, W. Li, X. Wan, P. Xie, T. Feng, K. Cheng, X. Geng *et al.*, "{SRNIC}: A scalable architecture for {RDMA}{NICs}," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 2023, pp. 1–14.

[6] J. Liu, J. Wu, S. P. Kini, P. Wyckoff, and D. K. Panda, "High performance rdma-based mpi implementation over infiniband," in *Proceedings of the 17th Annual International Conference on Supercomputing*, ser. ICS '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 295–304. [Online]. Available: https://doi.org/10.1145/782814.782855

[7] B. Alverson, E. Froese, L. Kaplan, and D. Roweth, "Cray xc series network," *Cray Inc., White Paper WP-Aries01-1112*, 2012.

[8] M. S. Birrittella, M. Debbage, R. Huggahalli, J. Kunz, T. Lovett, T. Rimmer, K. D. Underwood, and R. C. Zak, "Intel® omni-path architecture: Enabling scalable, high performance fabrics," in *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects*, 2015, pp. 1–9.

[9] R. K. Govindaraju, P. Hochschild, D. Grice, K. Gildea, R. Blackmore, C. A. Bender, C. Kim, P. Chaudhary, J. Goscinski, J. Herring, S. Martin, and J. Houston, "Architecture and early performance of the new ibm hps fabric and adapter," in *High Performance Computing - HiPC 2004*, L. Bougé and V. K. Prasanna, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 156–165.

[10] InfiniBand Trade Association, "Ibta specification volume 1 release 1.7," https://www.infinibandta.org/ibta-specification/, 2023, accessed: 2024-08-21.

[11] ——, "InfiniBand Trade Association," https://www.infinibandta.org, 1999, accessed: 2024-08-21.

[12] T. Hoefler, D. Roweth, K. Underwood, R. Alverson, M. Griswold, V. Tabatabaee, M. Kalkunte, S. Anubolu, S. Shen, M. McLaren, A. Kabbani, and S. Scott, "Data center ethernet and remote direct memory access: Issues at hyperscale," *Computer*, vol. 56, no. 7, pp. 67–77, 2023.

[13] InfiniBand Trade Association, "RDMA over Converged Ethernet (RoCE)," https://www.infinibandta.org/ibta-specification/, 2023, accessed: 2024-08-21, https://www.infinibandta.org/roce.

[14] C. Peterson, J. Sutton, and P. Wiley, "iwarp: a 100-mops, liw microprocessor for multicomputers," *IEEE Micro*, vol. 11, no. 3, pp. 26–29, 1991.

[15] M. J. Rashti and A. Afsahi, "10-gigabit iwarp ethernet: Comparative performance analysis with infiniband and myrinet-10g," in *2007 IEEE International Parallel and Distributed Processing Symposium*, 2007, pp. 1–8.

[16] W. Li, J. Liu, S. Wang, T. Zhang, S. Zou, J. Hu, W. Jiang, and J. Huang, "Survey on traffic management in data center network: From link layer to application layer," *IEEE Access*, vol. 9, pp. 38 427–38 456, 2021.

[17] C. Guo, H. Wu, Z. Deng, G. Soni, J. Ye, J. Padhye, and M. Lipshteyn, "Rdma over commodity ethernet at scale," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 202–215. [Online]. Available: https://doi.org/10.1145/2934872.2934908

[18] Y. Zhu, H. Eran, D. Firestone, C. Guo, M. Lipshteyn, Y. Liron, J. Padhye, S. Raindel, M. H. Yahia, and M. Zhang, "Congestion Control for Large-Scale RDMA Deployments," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM 2015. New York, NY, USA: Association for Computing Machinery, 2015, p. 523–536.

[19] Y. Zhang, Q. Meng, C. Hu, and F. Ren, "Revisiting congestion control for lossless ethernet," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024, pp. 131–148.

[20] J. P. Kenny, J. J. Wilke, C. D. Ulmer, G. M. Baker, S. Knight, and J. A. Friesen, "An Evaluation of Ethernet Performance for Scientific Workloads," in *2020 IEEE/ACM Innovating the Network for Data-Intensive Science (INDIS)*, 2020, pp. 57–67.

[21] Y. Li, H. Qi, G. Lu, F. Jin, Y. Guo, and X. Lu, "Understanding hot interconnects with an extensive benchmark survey," *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, vol. 2, no. 3, p. 100074, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772485922000618

[22] D. De Sensi, S. Di Girolamo, K. H. McMahon, D. Roweth, and T. Hoefler, "An in-depth analysis of the slingshot interconnect," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020, pp. 1–14.

[23] Ultra Ethernet Consortium, "Overview of and Motivation for the Forthcoming Ultra Ethernet Consortium Specification," 2023. [Online]. Available: https://ultraethernet.org/wp-content/uploads/sites/20/2023/10/23.07.12-UEC-1.0-Overview-FINAL-WITH-LOGO.pdf

[24] Huawei, "Hpc lossless ethernet and ai fabric network technical white paper - huawei enterprise," 2024. [Online]. Available: https://e.huawei.com/eu/material/enterprise/2698631dcfc14f468f7988bba2f5e722

[25] OEHI, "Welcome to haicgu cluster documentation¶." [Online]. Available: https://haicgu.github.io/

[26] Huawei, "Huawei ce8850 ce8850e data center switch datasheet - huawei enterprise," 2022. [Online]. Available: https://e.huawei.com/en/material/networking/dcswitch/0e2b9914aa134aeb9783fbd4d5b18137

[27] ——, "Huawei cloudengine 9855-32dq data center switch datasheet - huawei enterprise," 2022. [Online]. Available: https://e.huawei.com/en/material/enterprise/ddf8ccb37c844c09ba394453443dee34

[28] T. Hoefler and R. Belli, "Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '15. New York, NY, USA: Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2807591.2807644

[29] Y. Chen, R. Griffith, J. Liu, R. H. Katz, and A. D. Joseph, "Understanding tcp incast throughput collapse in datacenter networks," in *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*, ser. WREN '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 73–82. [Online]. Available: https://doi.org/10.1145/1592681.1592693

[30] S. Chunduri, S. Parker, P. Balaji, K. Harms, and K. Kumaran, "Characterization of mpi usage on a production supercomputer," in *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2018, pp. 386–400.

[31] A. Gibiansky, "Bringing hpc techniques to deep learning," https://andrew.gibiansky.com/blog/machine-learning/baidu-allreduce/, 2017, [Online; accessed 23-August-2024].

[32] A. Sergeev and M. D. Balso, "Horovod: fast and easy distributed deep learning in TensorFlow," *arXiv preprint arXiv:1802.05799*, 2018.

[33] D. Unat, I. Turimbetov, M. K. T. Issa, D. Sağbili, F. Vella, D. D. Sensi, and I. Ismayilov, "The landscape of gpu-centric communication," 2024. [Online]. Available: https://arxiv.org/abs/2409.09874

[34] D. D. Sensi, L. Pichetti, F. Vella, T. D. Matteis, Z. Ren, L. Fusco, M. Turisini, D. Cesarini, K. Lust, A. Trivedi, D. Roweth, F. Spiga, S. D. Girolamo, and T. Hoefler, "Exploring gpu-to-gpu communication: Insights into supercomputer interconnects," 2024. [Online]. Available: https://arxiv.org/abs/2408.14090