An Evaluation of Ethernet Performance for Scientific Workloads

Joseph P. Kenny, Jeremiah J. Wilke, Craig D. Ulmer, Gavin M. Baker, Samuel Knight, Jerrold A. Friesen
Scalable Modeling and Analysis
Sandia National Laboratories
Livermore, CA, USA
{jpkenny,jjwilke,cdulmer,gmbaker,sknigh,jafries}@sandia.gov

***Abstract*—Priority-based Flow Control (PFC), RDMA over Converged Ethernet (RoCE) and Enhanced Transmission Selection (ETS) are three enhancements to Ethernet networks which allow increased performance and may make Ethernet attractive for systems supporting a diverse scientific workload. We constructed a 96-node testbed cluster with a 100 Gb/s Ethernet network configured as a tapered fat tree. Tests representing important network operating conditions were completed and we provide an analysis of these performance results. RoCE running over a PFC-enabled network was found to significantly increase performance for both bandwidth-sensitive and latency-sensitive applications when compared to TCP. Additionally, a case study of interfering applications showed that ETS can prevent starvation of network traffic for latency-sensitive applications running on congested networks. We did not encounter any notable performance limitations for our Ethernet testbed, but we found that practical disadvantages still tip the balance towards traditional HPC networks unless a system design is driven by additional external requirements.**

***Index Terms*—interconnects, Ethernet, RoCE, RDMA, flow control, quality-of-servce (QoS)**

I. Introduction

Although Infiniband or proprietary networks dominate the top ten supercomputers and open science platforms in general, Ethernet is becoming more relevant in scientific computing. While lossless networks like Infiniband [1] have been preferred for high-performance computing (HPC) over "best-effort" networks like Ethernet, the evolution of Ethernet through features such as RDMA over Converged Ethernet (RoCE) and improved flow/congestion control have made it a more appealing option. Mellanox market shares have shifted heavily towards Ethernet and the trend is projected to continue [2]. Intel has abandoned its Omni-Path network and acquired the Ethernet-focused Barefoot Networks [3]. HPE Cray's new Slingshot interconnect [4], while a proprietary network, has Ethernet compatibility as a first-class design concern. Over 50% of the TOP500 systems are currently Ethernet [5], and the storage, hyperscale and hyperconverged markets are dominated by Ethernet.

Alongside the evolution of Ethernet, the workloads for research scientific computing are rapidly diversifying beyond traditional HPC to include broader data sciences. In particular, Ethernet is a requirement to support network emulation and cybersecurity applications within our lab [6]. With our datacenter now split between traditional high performance Infiniband and 100 Gb/s Ethernet networks, an obvious question for future procurements is whether Ethernet can effectively support our diversifying workload and thereby increase the flexibility of our systems and decrease our system deployment and maintenance overhead.

To support our technology evaluation work, we created a 96-node testbed with a tapered fat-tree network constructed from high-performance 100 Gb/s Ethernet hardware. In this work, we describe the network technologies that are available to improve performance on advanced Ethernet networks and evaluate the impact of these technologies on a small set of performance tests which has been tailored to exemplify the breadth of demands placed on networks by possible workloads. Overall, we demonstrate promising performance results for a set of HPC communication benchmarks.

II. Related Work

Several studies have looked at Ethernet benchmarks in the context of both HPC and other datacenter workloads. Work by Beck and Kagan showed latency improvements for a limited set of applications on older-generation 10 Gb/s Ethernet with RoCE [7] but did not have an HPC focus. Chuanxiong et. al. provided a detailed description of numerous issues encountered running RoCE over 40 Gb/s Ethernet at large scale in Microsoft datacenters, but also did not focus on HPC benchmarks [8]. Vienne et. al. is the most similar to our current work, providing a comprehensive comparison of performance for older QDR/FDR Infiniband and 10/40 Gb/s RoCE for both HPC and cloud computing workloads [9]. Priority-based Flow Control (PFC) and Enhanced Transmission Selection (ETS) were not considered and performance results were limited to a single switch. Beyond Ethernet, message-passing (MPI) implementations for HPC have been developed for other network interfaces, including Amazon Elastic Fabric Adapter [10].

Interest in RoCE performance has increased significantly in recent years. Low-level testing of the RoGUE congestion control and recovery mechanism was done on 10/40/100 Gb/s hardware [11]. Cheng et. al. performed low-level testing of

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

DOI 10.1109/INDIS51933.2020.00011

the Photonic Congestion Notification (PCN) scheme on a four-node testbed [12]. Mittal et. al. proposed a RoCE NIC design that tolerates packet losses and analyzed the performance of their design with simulations of synthetic traffic [13]. Shpiner et. al. performed incast tests including lossy RoCE configurations on a similar testbed but had more aggressive tapering in the upper level of their tree [14].

Quality-of-Service (QoS) for MPI applications has been studied extensively using Infiniband, but has not been thoroughly explored with Ethernet. Subramoni et. al. demonstrated latency improvements using QoS for interfering MPI traffic on a small Infiniband cluster [15]. Zhang et. al. examined the performance of interfering RDMA flows on Infiniband networks, and Patki et. al. explored running applications with isolated service levels over Infiniband [16], [17]. Simulation of Infiniband-like networks was used to study the impacts of QoS on HPC workloads in recent studies by Savoie et. al., Mubarak et. al, and two of the authors [18]–[20]. In a recent study of particular relevance to the current work, Balla et. al. used QoS to reduce RoCE latencies in the presence of interfering traffic, but did not consider HPC benchmarks.

III. Ethernet Performance Enhancements

Though Ethernet is uncommon in facilities supporting scientific research, three technologies have been introduced within the past decade which may enhance the performance of Ethernet for demanding workloads. Remote Direct Memory Access (RDMA) is the defining feature of a traditional HPC network. RDMA protocols allow communication traffic to bypass the operating system kernel, resulting in high performance, but these protocols rely on a lossless fabric which Ethernet does not provide. Priority-based Flow Control (PFC) was developed as an extension to earlier global flow control capabilities, allowing a fabric to pause flows belonging to selected priority levels in response to congestion [21]. With flow control providing a near-lossless Ethernet, RDMA operations can be performed by encapsulating Infiniband packets in Ethernet frames, and the RoCE standards specify how to do this. There are v1 and v2 RoCE standards (only v2 is routable) [22], [23]. There is also the potential for PFC-enabled networks to improve performance of the Transmission Control Protocol (TCP) more typically run over Ethernet. Lastly, Enhanced Transmission Selection (ETS) is a QoS approach which shares bandwidth between priority levels using a weighted round-robin algorithm [21]. ETS avoids the rigid bandwidth allocation or strict prioritization of other QoS approaches. Such approaches may not be desirable for interfering scientific applications which have equivalent importance but nonetheless can suffer from intermittent traffic "starvation" and corresponding tail latencies. As PFC, RoCE and ETS technologies mature, it is prudent to consider the possible performance impacts, costs and ease of adoption for Ethernet networks when specifying new system procurements.

IV. Ethernet Testbed Setup

The 100 Gb/s Ethernet testbed for this work, illustrated in Figure 1, was constructed as a two-level fat tree using eight 16-port Mellanox SN2100 leaf switches with a single 32-port Mellanox SN2700 core switch [24]. Each leaf switch was configured with links to 12 compute nodes and 4 links to the core switch for a 3:1 taper in the network and a total of 96 compute nodes. This is a significant level of tapering for a HPC network, but it is similar to what might be expected in a system designed for less demanding workloads and utilizing a typical top-of-rack switch layout. For stress testing of networking technologies this highly-tapered network also has the advantage of encouraging congestion in the upper level of the tree. RoCE v2 was used in this work unless explicitly stated and we did not configure end-to-end congestion control for the network, as link-level PFC proved sufficient. All MPI tests used Open MPI 4.0.4 [25]. Further details for the testbed hardware, firmware and software are provided in Table I.

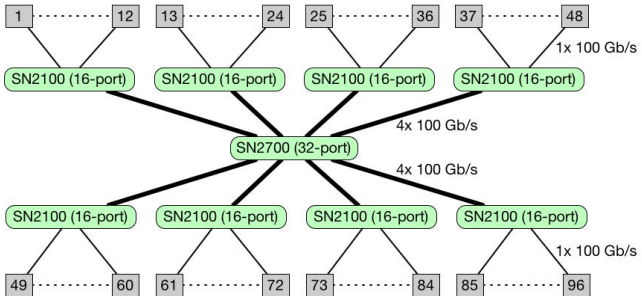

Fig. 1: Network architecture of 96-node 100 Gb/s Ethernet testbed.

V. Benchmarks

With the exception of a many-to-one incast benchmark, all of the tests used in this work utilized MPI for interprocess communication [26]. Though scientific computing is rapidly broadening beyond MPI, it is straightforward both to swap out network protocols and to control application priority levels, as required for this work, using the Open MPI implementation [25]. Point-to-point bandwidth and latency tests were obtained from the MVAPICH2 distribution [27]. A 3D halo exchange motif from the Ember library (`halo3d-26`) was used as a proxy for bandwidth-sensitive applications [28]. An FFT proxy application (`subcom3d-a2a`) was used to represent latency-sensitive applications [29]. Halo3D and FFT were run with one MPI rank per node, representing MPI+X parallelism. Random 48-node allocations were used consistently for Halo3D and FFT both in isolation and simultaneous execution, reflecting the disjoint allocations typically seen on production clusters. The High Performance Linpack (HPL) benchmark [30] was chosen to represent both MPI-only parallel applications and applications which may place more moderate demands on the network. HPL was run with one MPI rank on each of the 32 cores per node. Finally, a many-to-one incast test, which does not rely upon MPI, was performed to

CPU	2x Intel E5-2683 v4 (32 total cores)
OS/Kernel	CentOS Linux release 7.7.1908 / Linux 3.10.0-1062.9.1.el7.x86_64
NIC	Mellanox ConnectX-5 MT27800, hw_ver: 0x0, board_id: MT_0000000011
NIC Drivers/Firmware	MLNX_OFED_LINUX-4.6-1.0.1.1, fw_ver: 16.24.1000
Switch OS	Mellanox Onyx, version 3.8.2004 (3.7.1134 for SN2700)
MTU	4500
txqueuelen	1000 (default)
tcp_rmem	4096 87380 6291456 (default)
tcp_wmem	4096 16384 4194304 (default)
tcp_congestion_control	cubic (default)

TABLE I: Details of experimental setup. Default NIC/TCP parameters were used as performance did not improve in response to parameter tuning efforts.

examine the performance of the network links under extreme congestion. This configurable benchmark acts as a driver for low-level bandwidth test utilities (`iperf3` for TCP and `ib_write_bw` for RoCE), avoiding potential MPI overhead, and by varying the number of sources the bandwidth of the injected traffic and therefore the level of congestion can be tuned. This benchmark proved highly useful for diagnosing performance issues in earlier work tuning and testing RoCE configurations [31].

VI. Bandwidth and Latency Performance

Intraswitch point-to-point bandwidths and latencies were obtained on the testbed using MPI benchmarks as described in Section V. As seen in Figure 2, both RoCE v1 and RoCE v2 protocols achieve bandwidths above 97 Gb/s for large 4MB messages, very near the nominal 100 Gb/s rate, while TCP barely exceeds 30 Gb/s with or without flow control. Despite significant tuning efforts, we were not able to obtain the higher TCP single-stream bandwidths that have been reported for other 100 Gb/s Ethernet environments [32] and used default TCP settings. Similar performance gaps are seen for latencies in Figure 3, with TCP latencies ranging from 13-15 μs while both RoCE protocols are an order of magnitude lower, approaching 1 μs. While low TCP bandwidths may be overcome by using multiple streams, particularly as compute architectures continue to encourage increased parallelism, the low latencies provided by RoCE cannot be matched using TCP and will be a significant performance advantage for latency-sensitive applications. Though performance problems have been reported for some RoCE v1 implementations, we have seen no evidence of differing performance between v1 and v2 on our networks. We report results for RoCE v2 in the remainder of this work.

The incast benchmark was used to examine both the bandwidth available via multiple communication streams and the performance of highly congested links. The bandwidth attainable with multiple streams indicates the potential to ameliorate TCP performance in practical applications, and RoCE performance indicates the ability of PFC to avoid dropped packets and subsequent deterioration of performance under high levels of congestion. Figure 4 shows aggregate throughputs for this incast benchmark obtained by scanning the numbers of sending processes and source nodes that these processes are distributed over. Good performance is indicated by high and consistent aggregate throughput. For brevity, we report TCP results without PFC as the performance was slightly better than with PFC.

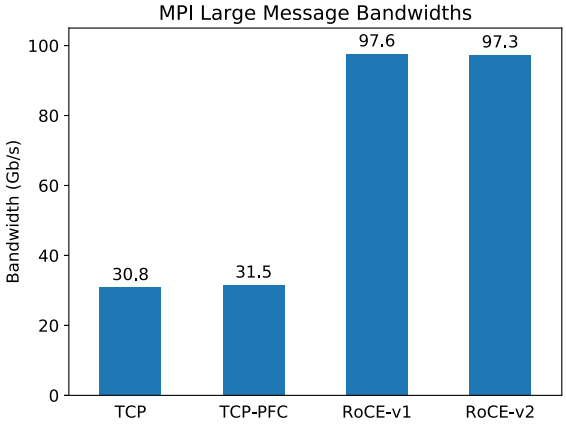

Fig. 2: MPI point-to-point bandwidths for message sizes of 4MB.

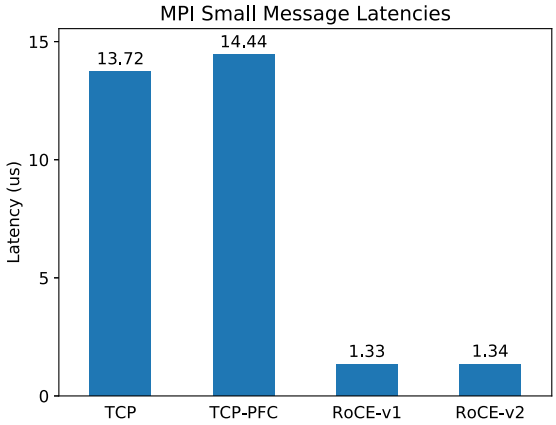

Fig. 3: MPI point-to-point latencies for zero-payload messages.

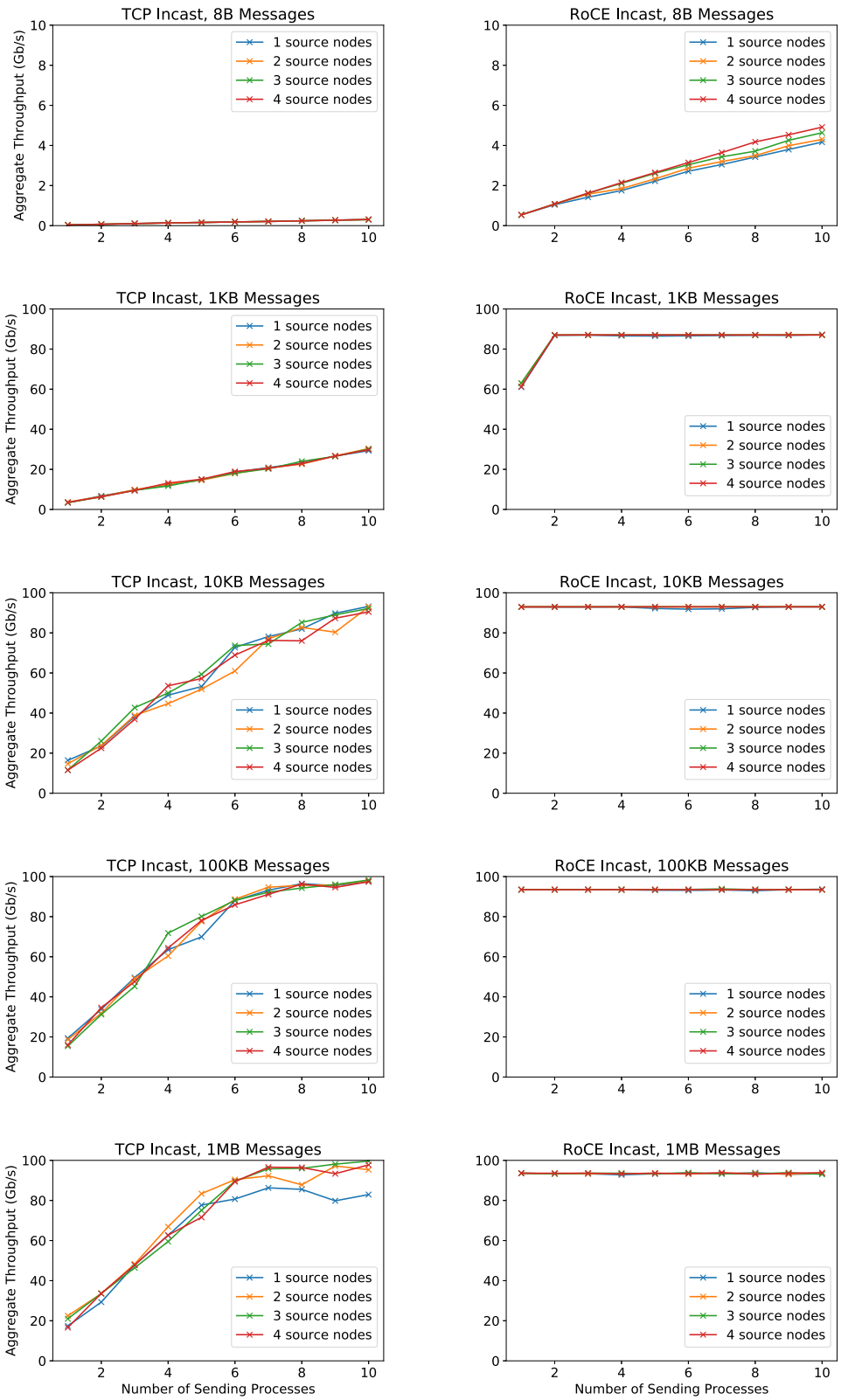

Fig. 4: Aggregate throughputs for incast testing of TCP and RoCE protocols. Note that a different plotting scale was used for 8B message size.

At message sizes of 10 KB or greater we see that TCP single stream bandwidths only reach about 20% of the 100 Gb/s nominal bandwidth, while bandwidths approach 100 Gb/s when a large number of sending processes are utilized. With message sizes smaller than 10 KB, bandwidth increases with the number of sending processes in a similar fashion but the maximum throughput is reduced. At the largest message size of 1MB performance variability is increased. When running the incast benchmark using RoCE protocols small message throughput is still low, but stable high performance, exceeding 90 Gb/s in most cases, is seen with message sizes ranging from 1 KB to 1MB over a wide range of sending process and source node numbers.

Taken as a whole, these small scale benchmarks suggest that RoCE will provide significant performance improvements for applications where high single-stream bandwidth or low latency are important. While TCP was able to approximately match the incast performance of RoCE under a very limited range of conditions, RoCE provided stable high throughput for this challenging benchmark over a broad range of parameters. Though an effective flow control implementation is critical for a network to successfully support RoCE protocols, these small-scale benchmarks did not reveal any instances in which PFC by itself improved performance when using TCP.

VII. Application Proxy Performance

With the knowledge that small scale bandwidth and latency benchmarks indicate significant potential performance advantages for Ethernet networks utilizing RoCE protocols, a small set of application proxies were run on the full Ethernet testbed. The bandwidth-intensive Halo3D proxy application is expected to benefit significantly when using RoCE, and, as seen in Figure 5, this is indeed the case. While the average application process spends 4.7 ms performing a Halo3D iteration using TCP, this time drops more than 80% to 0.9 ms using RoCE. Interestingly, this is the one example in our results where TCP with PFC provides a significant improvement over TCP without PFC. With PFC enabled in the networking hardware the average iteration time drops by 41% to 2.8 ms. FFT, a latency-sensitive application, likewise shows significant improvements when using RoCE (see Figure 6). While TCP both with and without PFC yields an average iteration time of 0.37 ms, RoCE drops this time by more than 80% to 0.07 ms. Figure 7 provides the percentage of theoretical peak FLOP/s attained for HPL. In this case the network has little impact on performance. The 71% efficiency for RoCE is not significantly better than the 64% found for TCP both with and without PFC enabled.

The summed switch counters for pause durations provided in Table III, which are not directly comparable between applications due to differing execution times, provide insight into these performance results. Unsurprisingly, the latency-sensitive FFT proxy does not create enough congestion to trigger any flow control pauses. With the Halo3D workload, for which congestion is expected, pauses occur almost entirely on the leaf switches. The leaf switches spend significant time pausing

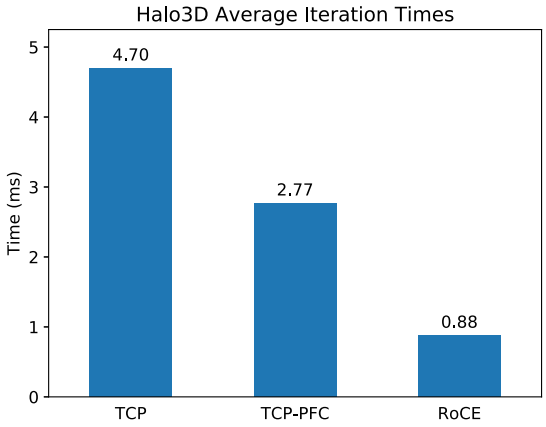

Fig. 5: Average iteration times for Halo3D.

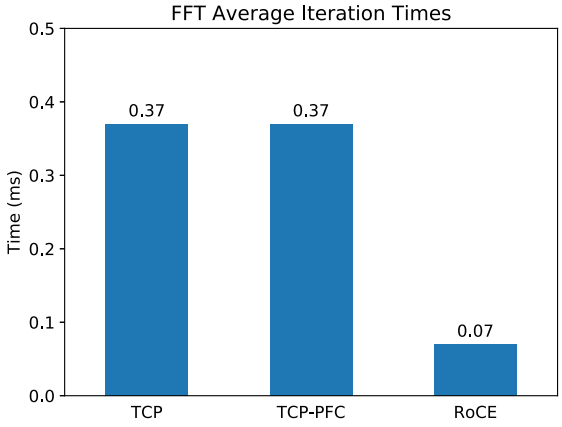

Fig. 6: Average iteration times for FFT.

transmission to the NICs, seen as receive pause duration, when TCP-PFC is used. This pause activity corresponds to the improved TCP iteration times for Halo3D when PFC is enabled, leading to the conclusion that PFC is able to better manage congestion on theses links, preventing TCP from backing off too aggressively. Somewhat counter-intuitively, the Halo3D pause durations actually drop when using the RoCE protocols, even though average network bandwidth increases significantly. We assume that this occurs because the nodes are able to process inbound messages faster due to the ability of RoCE to bypass the kernel. While HPL is not considered heavily taxing to networks, running 32 MPI ranks per node is sufficient to cause some congestion and flow control pauses in both directions throughout the network. It appears that the concurrent TCP streams from multiple MPI ranks per node fairly effectively utilize the available bandwidth, and enabling PFC for TCP does not result in any performance improvement. For similar reasons, moving to the RoCE protocol does not

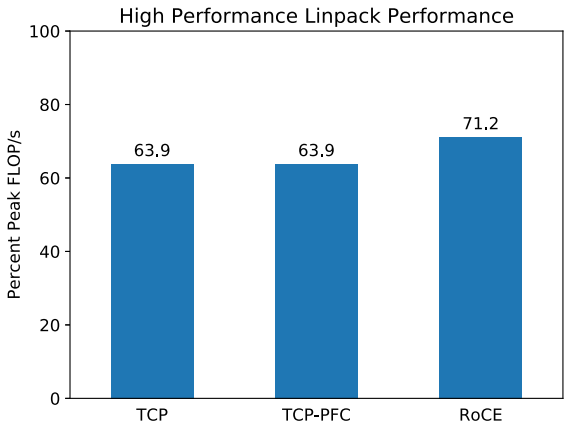

Fig. 7: High Performance Linpack percent of theoretical peak FLOP/s.

		Rx Pause Duration	Tx Pause Duration
Halo3D	TCP-PFC	6602760	0
	RoCE	120121	0
FFT	TCP-PFC	0	0
	RoCE	0	0
HPL	TCP-PFC	241264	174764
	RoCE	6929284	9404312

TABLE II: Summation of switch pause duration counters for application benchmarks run with PFC enabled. Due to differing execution times, the data between different applications are not directly comparable.

significantly increase performance; the increased bandwidth enabled by RoCE produces more congestion and an increase in flow control pauses, negating much of the benefit of RoCE for this case.

While clearly not an exhaustive study of application performance, these application proxy results provide a good illustration of the advantages that some workloads may see with RoCE/PFC. For both Halo3D and FFT, respectively bandwidth-sensitive and latency-sensitive applications, RoCE provides a greater than 80% improvement in average iteration time over TCP. However, the much smaller improvement when moving to RoCE for HPL reminds us that these performance improvements should be considered best-case scenarios for particularly well-suited workloads.

VIII. Managing Application Interference with Enhanced Transmission Selection

While all of the tests reported thus far have been proxies for single applications running in isolation, production computing clusters typically run a number of simultaneous jobs with varying resource demands. QoS supports differentiation between network traffic, allowing for resources to be flexibly allocated. In previous work, some of the authors have demonstrated via simulation that significant performance can be gained for latency-sensitive applications running on a congested network by allocating bandwidth using a weighted-round-robin algorithm analogous to ETS. While the latency-sensitive application does not require significant bandwidth, with QoS configured the network hardware has dedicated buffer resources for the priority level that the application is assigned to. These dedicated buffer resources allow the time-sensitive traffic to bypass congestion, preventing high-bandwidth applications from blocking the latency-sensitive application's traffic.

Using the Ethernet testbed constructed for this work, we examined the efficacy of a real-world ETS implementation for this type of workload. For this experiment we used the same Halo3D and FFT proxy applications which were examined in Section VII. Since these applications were chosen to represent bandwidth-sensitive (Halo3D) and latency-sensitive (FFT) applications, they are well suited to reproduce the desired application interference. Each proxy was run on a randomly assigned set of 48 nodes and given a 50% bandwidth weight, matching bandwidth weight to compute resources. FFT was launched 10 seconds after a longer running Halo3D computation.

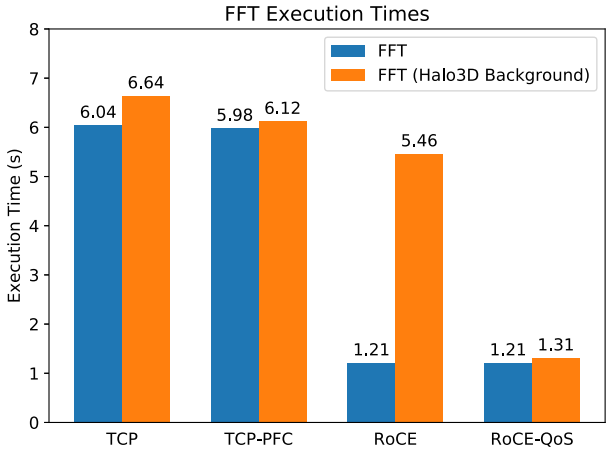

Fig. 8: Total execution times for FFT both with and without Halo3D in background.

Figure 8 compares total execution times when running this QoS experiment using different network capabilities. Interestingly, the poor bandwidth performance of TCP in effect throttles Halo3D to such an extent that it does not significantly interfere with FFT. FFT takes 6.0 seconds to run in the absence of Halo3D and only slows to 6.6 seconds when sharing the network, while the equivalent slow down for RoCE is from 1.2 to 5.5 seconds – a factor greater than four. As seen in the individual results (Figure 5), Halo3D execution times with TCP drop significantly when PFC is enabled. However, FFT performance is not significantly impacted by Halo3D background traffic even when PFC is enabled, indicating that the network remains relatively uncongested beyond the switch to NIC links dedicated to Halo3D traffic.

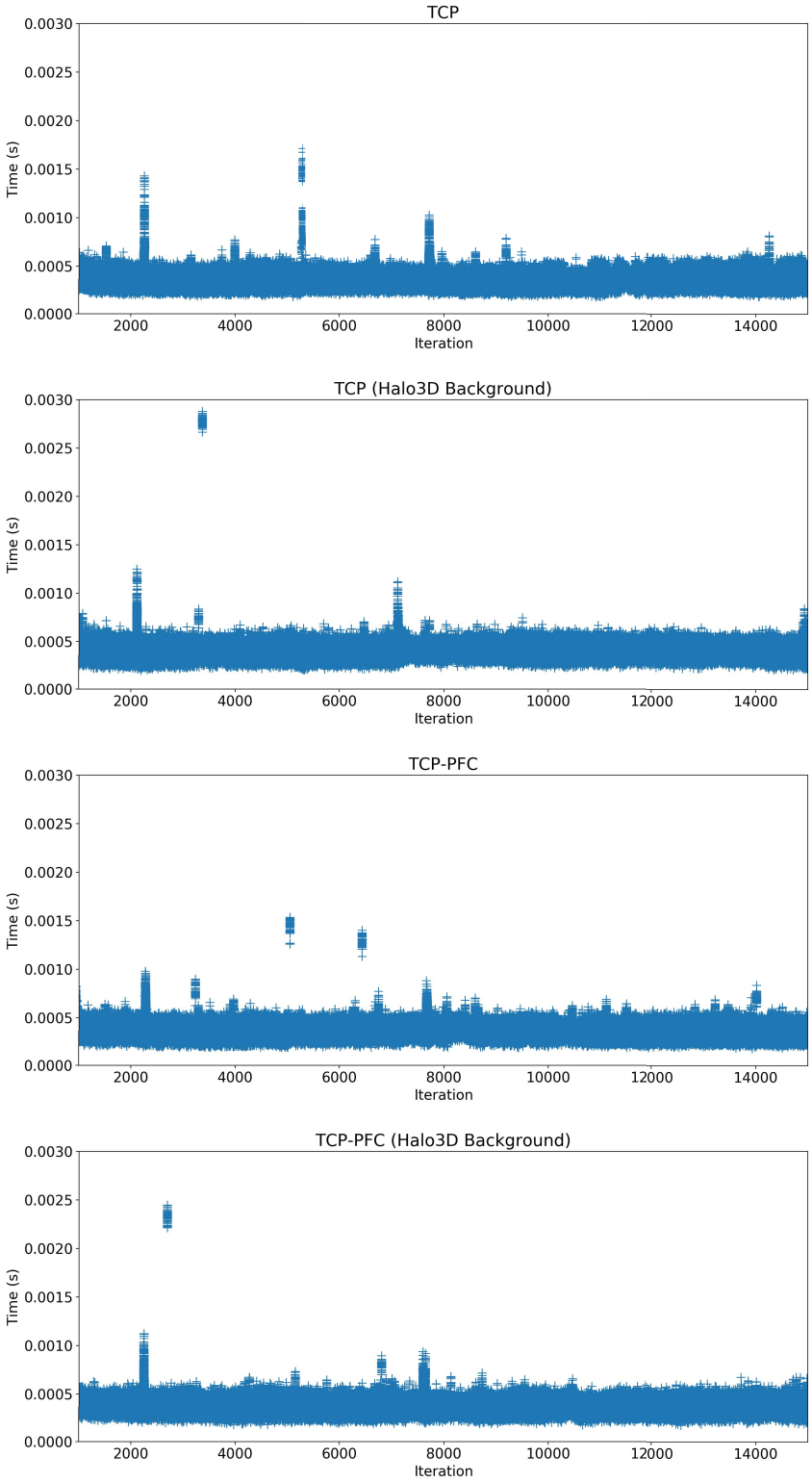

Fig. 9: Individual per-node iteration times for FFT running with TCP protocols with and without interference from Halo3D background traffic.

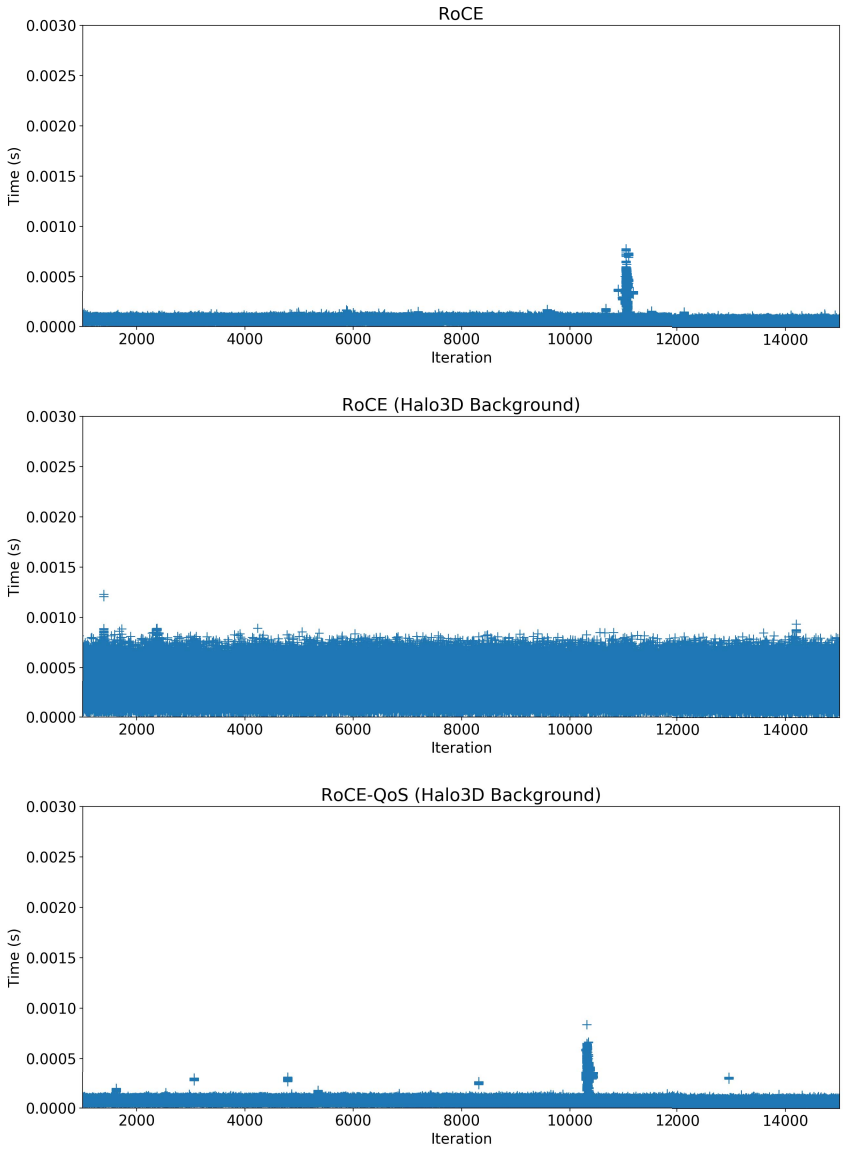

Fig. 10: Individual per-node iteration times for FFT running with RoCE protocols with and without interference from Halo3D background traffic.

Moving from TCP to the RoCE protocol, when Halo3D background traffic is included FFT is not able to benefit from the vastly improved RoCE latency and only a modest improvement in total execution time is observed, unlike the case of running FFT in isolation. This is where ETS demonstrates significant potential. Enabling QoS for RoCE drops the FFT execution time to just 1.3 seconds, more than a 75% reduction of the 5.5 seconds observed for RoCE without QoS.

These effects can be further understood by examining plots of the individual per-node iteration times for FFT. Iteration times for TCP protocols are show in Figure 9. Results for TCP and TCP-PFC are qualitatively very similar, both with and without Halo3D background traffic, again demonstrating that Halo3D does not saturate the network when using TCP. Iteration times are clustered around 0.5 ms and a small number of iterations show marked increase in performance variability.

Switching to the RoCE protocol (Figure 10), FFT without Halo3D background traffic has iteration times tightly clustered around the 0.07 ms average, with the exception of a large spike in variability appearing between 10 and 12 thousand iterations. Adding background traffic without QoS, many fast iterations remain but the spread in iteration times is drastically increased. Compared to TCP, many nodes have lower iteration times but the overall execution time does not improve significantly since the all-to-all operations in FFT act as barriers. When QoS is enabled for RoCE and the interfering applications are given separate priority levels, FFT has dedicated buffer space and experiences much less network delay. The overwhelming majority of iterations now complete in under 0.25 ms, accounting for the significant drop in total execution time to 1.3 seconds. The spike in performance variability between 10 and 12 thousand iterations appears again along with intermittent small slow downs on a subset of nodes. RoCE-QoS results without background traffic (not shown) are very similar to those obtained with background traffic, including these intermittent small slow downs. The QoS-enabled network, even with significant congestion, is able to very nearly reproduce the RoCE performance of FFT in isolation. This experiment demonstrates the significant performance gains latency-sensitive applications can experience when ETS is used to differentiate their traffic from high-bandwidth workloads.

Examining the summed switch performance counters for this experiment, shown in Table III, revealed unexpected behavior. The counter-intuitive drop in pauses between TCP-PFC and RoCE/RoCE-QoS was discussed in Section VII. For all three network options priority 1 counters show pause packets and durations which are nearly identical to priority 0, even though TCP-PFC and RoCE in fact have no traffic with priority 1. There does not appear to be any significant amount of time when only one priority is paused, though the PFC standard clearly "allows link flow control to be performed on a per-priority basis" [21]. It is not altogether surprising that a PFC implementation would behave in this manner. By default the buffers in the Mellanox switches have a mix of reserved and shared space; if a link has nearly exhausted its resources for one priority then traffic from other priorities may also be at risk of being dropped. Consequently, the observed performance improvements for FFT in this example are most likely attributable to the dedicated buffer resources allocated to its priority level and not to more optimal flow control behavior. Yet, we have encountered switches where global pause performs poorly in comparison to PFC, so we recommend using PFC [31].

IX. Conclusions

The bandwidth and latency tests presented in this work demonstrate that PFC and RoCE, as implemented in modern high-performance Ethernet networks, enable performance on par with traditional HPC networks. Evaluating these technologies on a tapered 96-node fat tree we found that application proxies representing both bandwidth-sensitive (Halo3D) and latency-sensitive (FFT) workloads were able to realize significant performance benefits, while HPL saw much less benefit. Critically, with this particular testbed configuration, PFC appeared to work with no packet drops and corresponding loss in throughput at the scale of these tests and allowed RoCE to perform well even under considerable congestion. Earlier RoCE evaluation efforts within our facility were not as successful, likely indicating an ecosystem which is still maturing [31].

ETS does allow differentiation between equal priority jobs running on a cluster, unlike bandwidth shaping or strict prioritization which are much more appropriate for managing traffic in more typical datacenters (e.g. storage or multimedia streams). We demonstrated that, in the specific case of a latency-sensitive application proxy (FFT) running on a congested network, assigning applications to separate priority levels in the ETS scheme can yield significant performance benefits and reduce starvation of network traffic due to heavy bandwidth consumers. While this case study demonstrates significant potential performance benefits, the practical value of incorporating ETS configuration into system management infrastructure would largely depend on the exact makeup of an installation's workload.

The results of this work demonstrate that, at least up to the scale of our testbed, modern Ethernet networks can yield competitive performance. However, Ethernet networks are not without disadvantages. In practice we have found that, while traditional HPC networks which use credit-based flow control provide more or less plug-and-play performance at moderate scales, the advanced capabilities which support high performance on Ethernet are challenging to configure and may lack feature maturity [31]. While the performance results are promising, EDR Infiniband with less configuration complexity has been significantly cheaper per port than 100 GB/s Ethernet in recent procurements [33]. Still, where particular device support or user demands shift requirements, Ethernet seems viable for new general purpose scientific computing clusters.

	Rx0 Pause Packets	Rx0 Pause Duration	Rx1 Pause Packets	Rx1 Pause Duration
TCP-PFC	1580102	11477936	1581330	11488489
RoCE	14312	64272	14312	64270
RoCE-QoS	23750	126279	23750	126279

TABLE III: Summation of switch flow control counters for FFT (priority 1 for RoCE-QoS and priority 0 otherwise) running with Halo3D (always priority 0) in background.

REFERENCES

[1] Infiniband Trade Association. [Online]. Available: http://www.infinibandta.org

[2] Mellanox. (2020) Mellanox corporate update. [Online]. Available: https://www.mellanox.com/related-docs/company/MLNX_Corporate_Deck.pdf

[3] T. P. Morgan. (2019) Intel goes Barefoot as it leaves the Omni-Path. [Online]. Available: https://www.nextplatform.com/2019/08/01/intel-goes-barefoot-as-it-leaves-the-omni-path

[4] HPE Cray. (2019) Slingshot: The interconnect for the exascale era. [Online]. Available: https://www.cray.com/sites/default/files/Slingshot-The-Interconnect-for-the-Exascale-Era.pdf

[5] TOP500.org. (2020) TOP500 List. [Online]. Available: https://www.top500.org/lists/top500/

[6] J. Floren, J. Friesen, C. Ulmer, and S. T. Jones, "A reference architecture for emulytics clusters," in *Sandia Report*, vol. SAND2009-5574, 2017.

[7] M. Beck and M. Kagan, "Performance evaluation of the RDMA over Ethernet (RoCE) standard in enterprise data centers infrastructure," in *Proceedings of the 3rd Workshop on Data Center-Converged and Virtual Ethernet Switching*. International Teletraffic Congress, 2011, pp. 9–15.

[8] 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*. ACM, 2016, pp. 202–215.

[9] J. Vienne, J. Chen, M. Wasi-Ur-Rahman, N. S. Islam, H. Subramoni, and D. K. Panda, "Performance analysis and evaluation of Infiniband FDR and 40GigE RoCE on HPC and cloud computing systems," in *2012 IEEE 20th Annual Symposium on High-Performance Interconnects*. IEEE, 2012, pp. 48–55.

[10] S. Chakraborty, S. Xu, H. Subramoni, and D. Panda, "Designing scalable and high-performance MPI libraries on Amazon Elastic Fabric Adapter," in *2019 IEEE Symposium on High-Performance Interconnects (HOTI)*. IEEE, 2019, pp. 40–44.

[11] Y. Le, B. Stephens, A. Singhvi, A. Akella, and M. M. Swift, "RoGUE: RDMA over Generic Unconverged Ethernet," in *Proceedings of the ACM Symposium on Cloud Computing*, 2018, pp. 225–236.

[12] W. Cheng, K. Qian, W. Jiang, T. Zhang, and F. Ren, "Re-architecting congestion management in lossless Ethernet," in *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, 2020, pp. 19–36.

[13] R. Mittal, A. Shpiner, A. Panda, E. Zahavi, A. Krishnamurthy, S. Ratnasamy, and S. Shenker, "Revisiting network support for RDMA," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 313–326.

[14] A. Shpiner, E. Zahavi, O. Dahley, A. Barnea, R. Damsker, G. Yekelis, M. Zus, E. Kuta, and D. Baram, "RoCE rocks without PFC: Detailed evaluation," in *Proceedings of the Workshop on Kernel-Bypass Networks*, 2017, pp. 25–30.

[15] H. Subramoni, P. Lai, and D. K. Panda, "Designing QoS aware MPI for InfiniBand - Technical Report," 2009.

[16] Y. Zhang, J. Gu, Y. Lee, M. Chowdhury, and K. G. Shin, "Performance isolation anomalies in RDMA," in *Proceedings of the Workshop on Kernel-Bypass Networks*, 2017, pp. 43–48.

[17] T. Patki, E. Ates, A. Coskun, and J. Thiagarajan, "Understanding simultaneous impact of network QoS and power on HPC application performance," in *Computational Reproducibility at Exascale (CRE'18), Supercomputing Workshop*, 2018.

[18] L. Savoie, D. K. Lowenthal, B. R. D. Supinski, and K. Mohror, "A study of network Quality of Service in many-core MPI applications," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2018, pp. 1313–1322.

[19] M. Mubarak *et al.*, "Evaluating Quality of Service traffic classes on the Megafly network," 2019.

[20] J. J. Wilke and J. P. Kenny, "Opportunities and limitations of Quality-of-Service (QoS) in Message Passing (MPI) applications on adaptively routed Dragonfly and Fat Tree networks," in *2020 IEEE Conference on Cluster Computing (CLUSTER)*, 2020.

[21] IEEE Computer Society, "IEEE Std 802.1Q-2018," 2018.

[22] Infiniband Trade Association and others, "InfiniBand Architecture Specification Release 1.2.1 Annex A16: RoCE," 2010.

[23] ——, "InfiniBand Architecture Specification Release 1.2.1 Annex A17: RoCEv2," 2014.

[24] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE transactions on Computers*, vol. 100, no. 10, pp. 892–901, 1985.

[25] (2020) Open MPI Home Page. [Online]. Available: https://www.open-mpi.org/

[26] (2015) MPI: A Message-Passing Interface Standard; Version 3.1. [Online]. Available: http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf

[27] (2020) MVAPICH Home Page. [Online]. Available: https://http://mvapich.cse.ohio-state.edu/

[28] (2020) Ember Element Library. [Online]. Available: http://sst-simulator.org/SSTPages/SSTElementEmber/

[29] (2020) Chatterbug Repository. [Online]. Available: https://github.com/hpcgroup/chatterbug

[30] (2020) Netlib HPL Webpage. [Online]. Available: https://www.netlib.org/benchmark/hpl/

[31] J. P. Kenny and C. D. Ulmer, "RoCE: Promising technology for Ethernet as a high performance networking fabric," in *Sandia Report*, vol. SAND2019-13444, 2019.

[32] N. Hanford and B. Tierney, "Recent Linux TCP updates, and how to tune your 100G host," 2016.

[33] Colfax Direct — Infiniband and Ethernet. [Online]. Available: https://www.colfaxdirect.com/store/pc/home.asp

Appendix: Artifact Description/Artifact Evaluation

SUMMARY OF THE EXPERIMENTS REPORTED

We ran point-to-point bandwidth and latency, incast, High Performance Linpack, 3D halo exchange proxy, and FFT proxy benchmarks on an Ethernet testbed as described in the paper and our data artifact. The data artifact includes relevant switch settings and arguments used to run the benchmarks.

ARTIFACT AVAILABILITY

Software Artifact Availability: There are no author-created software artifacts.

Hardware Artifact Availability: There are no author-created hardware artifacts.

Data Artifact Availability: All author-created data artifacts are maintained in a public repository under an OSI-approved license.

Proprietary Artifacts: None of the associated artifacts, author-created or otherwise, are proprietary.

List of URLs and/or DOIs where artifacts are available:
`https://github.com/jpkenny/ethernet-performance-data`

BASELINE EXPERIMENTAL SETUP, AND MODIFICATIONS MADE FOR THE PAPER

Relevant hardware details: Intel E5-2683 v4, Mellanox ConnectX-5, Mellanox SN2100, Mellanox SN2700

Operating systems and versions: CentOS Linux release 7.7.1908 / Linux 3.10.0-1062.9.1.el7.x86 64

Compilers and versions: gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39)

Applications and versions: osu_bw, osu_lat, HPL v2.2, halo3d-26, subcom3d-a2a, custom incast script

Libraries and versions: Open MPI 4.0.4, intel-mkl-2017.2.174

Key algorithms: LU factorization, halo exchange, FFT