RDMA over Ethernet - A Preliminary Study

Hari Subramoni, Ping Lai, Miao Luo and Dhabaleswar K. Panda

Department of Computer Science and Engineering, The Ohio State University
{subramon, laipi, luom, panda}@cse.ohio-state.edu

Abstract

*Though convergence has been a buzzword in the networking industry for sometime now, no vendor has successfully brought out a solution which combines the ubiquitous nature of Ethernet with the low latency and high performance capabilities that InfiniBand offers. Most of the overlay protocols introduced in the past have had to bear with some form of performance trade off or overhead. Recent advances in InfiniBand interconnect technology has allowed vendors to come out with a new model for network convergence - RDMA over Ethernet (*RDMAoE*). In this model, the IB packets are encapsulated into Ethernet frames thereby allowing us to transmit them seamlessly over an Ethernet network. The job of translating InfiniBand addresses to Ethernet addresses and back is taken care of by the InfiniBand HCA. This model, allows end users access to large computational clusters through the use of ubiquitous Ethernet interconnect technology while retaining the high performance, low latency guarantees that InfiniBand provides. In this paper, we present a detailed evaluation and analysis of the new RDMAoE protocol as opposed to the earlier overlay protocols as well as native-IB and socket based implementations. Through these evaluations, we also look at whether RDMAoE brings us closer the eventual goal of network convergence. The experimental results obtained with verbs, MPI, application and data center level evaluations show that RDMAoE is capable of providing performance comparable to Native-IB based applications on a standard 10GigE network.*

This research is supported in part by DOE grants #DE-FC02-06ER25749 and #DE-FC02-06ER25755; NSF grants #CNS-0403342, #CCF-0702675 and #CCF-0833169; grant from Wright Center for Innovation #WCI04-010-OSU-0; and equipment donations from Intel, Mellanox and Obsidian.

I. Introduction

Ever increasing demands for High Performance Computing systems and high performance to cost ratios have led to the growth of commodity clusters. Modern interconnects such as InfiniBand (IB)[8] and 10-Gigabit Ethernet (10GigE)[6] are widely deployed in such clusters to enhance the performance. These clusters are further connected to fulfill the needs for more powerful computing and storage environments, leading to the deployment of cluster-of-clusters or wide-area-networked clusters. In this context, it is critical to design efficient inter-cluster communication.

Multiple mechanisms can be used to deliver traffic across clusters, as presented in Figure 1. For clusters equipped with IB, it is natural to use IB switches or routers to connect the clusters and extend intra-cluster native IB traffic beyond one cluster. However, many existing infrastructures are equipped with only Ethernet switches or IP routers due to technology and cost concerns. Therefore, 10GigE inter cluster interconnects are still popularly used in many clusters because of their ubiquitous compatibility, even though end-to-end latency and throughput lag behind the performance given by IB. This case refers to TCP/IP in Figure 1. On the other hand, IB also supports such inter-operability with IP using some over-layered protocols such as IPoIB (IP over IB) [1], while at the same time losing some of the superior native IB performance. This is corresponding to IPoIB in Figure 1. As we have seen, all of these options have to trade off between the cost and performance, which leads to a question of whether it is possible to create a protocol capable of using the ubiquitous nature of Ethernet and delivering the high performance guaranteed as well.

The ConnectX [12] series of InfiniBand interconnects have recently introduced the capability to allow InfiniBand Traffic to flow over Ethernet, thereby allowing native IB traffic to flow seamlessly over an Ethernet link (RDMAoE)

[13]. This mechanism is named as RDMAoE, while the existing IB mechanism is named as native IB in Figure 1. The advantage of RDMAoE is that we can now leverage the high performance offered by native IB communication in an ubiquitous environment like Ethernet. The IB packets are encapsulated into Ethernet frames and sent out of the interconnect so that the intermediate Ethernet based network elements are able to transmit the packet to the intended destination. The job of translating the IB destination LID to an Ethernet understandable machine (MAC) address is handled in part by the End application/MPI stack and the InfiniBand HCA. In this scenario, clusters can use the existing Ethernet switch or router deployment while using the verbs level IB communication.

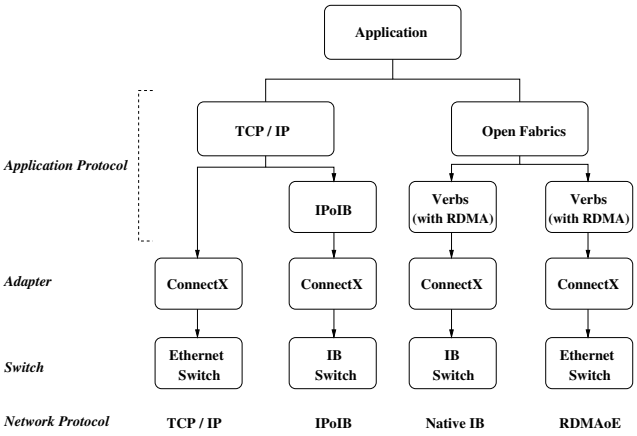

Figure 1. Overall Framework for Network Communication with ConnectX adapter

In this work, we target on evaluating and analyzing the performance characteristics of the aforementioned communication mechanisms. We basically want to answer the following questions through this research:

- How do the different communication protocols stack up against each other as far as performance is concerned?

and more broadly

- Does RDMAoE bring us a step closer to the goal of network convergence?

We first use a set of protocol level benchmarks to compare their very basic performance. After that, we characterize their performance using MPI [14] benchmarks (on both micro-benchmark level and application benchmark level), file system benchmarks and data center applications (i.e., FTP service). From the results, we observe that RDMAoE offers very close performance to native IB on verbs level, MPI benchmarks level and data center application level when transparently delivering traffic over Ethernet switch. However, it requires some amount of CPU time to encode/decode the packets, which results in high CPU utilization.

The remainder of this paper is organized as follows: Section II gives an overview of IB and RDMAoE. We evaluate and analyze the performance in various scenarios in Section III, describe the related work in Section IV, and summarize the conclusions and future work in Section V.

II. InfiniBand and RDMAoE

In this section we give a brief introduction on InfiniBand and RDMAoE.

A. InfiniBand

InfiniBand Architecture [8] is an industry standard that defines a System Area Network (SAN) to design clusters offering low latency and high bandwidth. A typical IBA cluster consists of switched serial links for interconnecting both the processing nodes and the I/O nodes. IBA supports two types of communication semantics: Channel Semantics (Send-Receive communication model) and Memory Semantics (RDMA communication model). Remote Direct Memory Access (RDMA) [21] operations allow processes to access the memory of a remote node process without the remote node CPU intervention. These operations are transparent at the remote end since they do not involve the remote CPU in the communication. Increasing number of InfiniBand clusters are currently being deployed in several High End Computing scenarios including high performance computing systems, web and Internet datacenters, etc.

The popular TCP/IP network protocol stack can be adapted for use with InfiniBand by the IP over IB (IPoIB) driver [11]. IPoIB is a Linux kernel module than enables InfiniBand hardware devices to encapsulate IP packets into IB datagram or connected transport services. When IPoIB is applied, an InfiniBand device is assigned an IP address and accessed just like any regular TCP/IP hardware device.

B. RDMA over Ethernet (RDMAoE)

RDMAoE is a new protocol that allows us to perform native IB communication seamlessly over lossless Ethernet links. RDMAoE packets are encapsulated into standard Ethernet frames with an IEEE assigned Ethertype, a Global Routing Header (GRH), unmodified InfiniBand transport headers and payload.

In RDMAoE, Ethernet management services are used instead of IB subnet management services. In Ethernet world, nodes are commonly referred to by applications through IP addresses. RDMAoE can encode the IP addresses of the corresponding Ethernet port into its Global

Identifier (GID), and makes use of the IP stack to bind a destination address to the corresponding netdevice and to obtain its L2 MAC addresses. In this way, the IB Verbs API need not be modified. When using RDMAoE ports, address handles are required to contain GIDs and the L2 address fields. The Ethernet L2 information is then obtained by using vendor-specific driver calls. Thus, the RDMAoE ports are functionally equivalent to regular IB ports from the IB stack perspective. Another interesting feature of this mode is that, for any socket based application, it is functionally equivalent to a normal 10 GigE port. Thus both socket based and IB verbs based applications can run over the ConnectX card in RDMAoE mode at the same time. As there is no Subnet Agent (SA) in RDMAoE, the connection management code is modified to fill the necessary path record attributes locally before sending connection management packets. Similarly, the connection manager provides to the user the required address handle attributes when processing Secure Inter-Domain Routing (SIDR) requests and joining multicast groups.

As in InfiniBand mode, most of the overhead of packet processing is taken care of by the ConnectX HCA. While TCP Segmentation Offload (TSO) is supported by the ConnectX HCA in hardware, Large Receive Offload (LRO) is a pure software feature used by the CX drivers in Linux. In RDMAoE mode, the ConnectX HCA only takes care of flow control at the transport level, leaving the link level flow control to the ones adopted by Ethernet. The HCA also performs offloading of TCP/UDP/IP checksum while operating as a pure Ethernet device. [1]

III. Performance Evaluation

In this section, we present the experimental results comparing the performance of the four communication mechanisms over IB ConnectX NIC's, inter connected using an IB or Ethernet switch. As the initial step, we measured the latency with two nodes connected back-to-back, in order to present the overhead of packet encapsulation in RDMAoE. We first measured the pure verbs level latency and CPU utilization. This provides the baseline reference for the performance of higher level protocols. We then compare their performance using OSU MPI level micro-benchmarks [20], IMB collective benchmarks [9], and NAS benchmark [3]. Finally we use FTP application to characterize their performance in a data center like environment.

[1]This protocol is undergoing changes and the description of RDMAoE is based on the protocol as it stands during the camera-ready preparation time (August '09).

A. Experimental Setup

Our experimental setup is as shown in Figure 2. Each node in our setup has the Intel Nehalem series of processors with Dual quad-core processor nodes operating at 2.40 GHz with 12 GB RAM and a PCIe 2.0 interface. The latest ConnectX DDR HCA was used in each of the compute nodes. Depending on the mode in which the ConnectX HCA's were configured - Native IB or RDMAoE, the switch used was either a 24 port DDR Mellanox IB switch or a 24 port Fulcrum Focal Point 10GigE switch respectively. Both the switches and the HCA's used CX4 [7] type interfaces. While ConnectX firmware version 2.6.0 and OFED-1.4.1 were used for Native IB and IPoIB related experiments, an experimental version of the ConnectX firmware and a pre-release version of OFED based out of OFED-1.5 was used for all the RDMAoE and TCP/IP based ones.

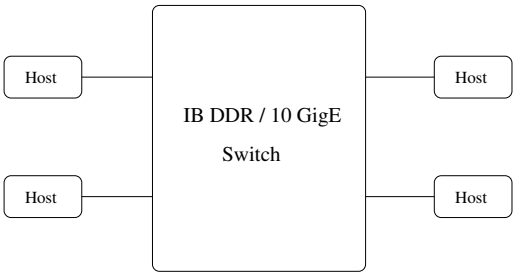

Figure 2. Experimental Setup

B. Verbs Level Performance

In this section, we use the IB verbs-level tests (*perftests*) provided with the OFED software stack to evaluate the performance of the basic IB protocols. To measure the performance of RDMAoE, we modified this benchmark to make it work compatibly with RDMAoE stack. Moreover, nttcp [4] benchmark is used to measure the performance of TCP/IP and IPoIB protocols.

First, we connect the machines in a back-to-back manner to find out the basic latency performance of the various operating modes. The results are shown in Figures 3 (a), (b) and (c) for small, medium and large messages, respectively. We see that the TCP-based protocols (either IPoIB or TCP/IP) do not perform well for small and medium message size, while RDMAoE has good performance as compared to the native IB. This means that the overhead of encapsulating the IB packets is marginal. Further, in order to make the scenario more realistic, we introduced a switch between two nodes. We performed the verbs level latency tests again to assess the overhead caused due to adding

the switch into the topology. Figures 4 (a), (b) and (c) present the corresponding latency performance. For small to medium sized messages, while the Mellanox DDR IB switch adds an average latency of *0.20 us* to *0.30 us*, the Fulcrum 10 GigE switch adds an average latency of *0.50 us* to *1.00 us*. In both the experimental scenarios, we see that RDMAoE yields the similar performance as native IB for small and medium messages. Therefore, it is expected to provide acceptable compromise between the performance and the supplies of existing switches.

We also measured the CPU utilization when the benchmark is running. The results are presented in Figure 5. The results shown here are normalized to be per core CPU utilization. Since RDMAoE targets on achieving very low latency communication, which is more notable when the message size is small, we only show the performance for small messages. As expected, native IB has very low CPU utilization, because it does not require much involvement from host processor to transmit packets. The reason RDMAoE shows higher CPU utilization could be due factors like the smaller MTU, which we plan to investigate in greater detail in the future. The amount of data that can be sent in one packet while the HCA is operating in RDMAoE mode is limited by the MTU of the lower level ethernet device which was set to 1500 bytes during our experiments. Apart from this, there is also the overhead of the IB level headers that needs to be encapsulated into the ethernet frame to consider. On the other hand, TCP-based protocols (TCP/IP and IPoIB) also need to occupy some CPU time for the TCP stack processing. IPoIB is highly optimized in the current OFED version, so it shows lower utilization here. Moreover, for both of them, the CPU utilization decreases as the message size increases. This is because that the TCP stack processing overhead is amortized by the large message transmission time.

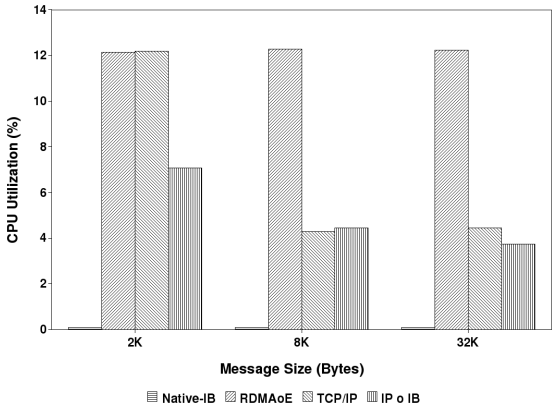

Figure 5. CPU Utilization Comparison

C. MPI Level Performance

In this section, we use the MPI as the higher level protocol to show the performance difference of the four lower level protocols. We measure the results using micro-benchmark, the collectives benchmark and NAS application benchmark.

MVAPICH [15] compiled with IB channel is utilized as the MPI model that directly runs over IB and RDMAoE, while the one compiled with TCP channel is utilized as the model running over TCP.

1) Micro Benchmark Level Performance: We utilize the OSU MPI Benchmark (OMB) [20] to characterize the performance comparison.

Figures 6 (a), (b) and (c) compare the bandwidth performance, including one-directional bandwidth, one-directional multi-pair communication bandwidth and bi-directional bandwidth. Similar to the verbs level results, MVAPICH over native IB has the best performance and MVAPICH over TCP/IP or IPoIB has the worst performance. The performance of RDMAoE falls in between, having about 25% to 30% degradation as compared to native IB. Using multiple communication pairs can mitigate the overhead of TCP stack processing, but cannot help much for IB. This is why there is improvement for TCP/IP and IPoIB in multi-pair bandwidth experiments, while no improvement seen in native IB and RDMAoE cases.

Figure 7 shows the MPI level latency performance. Similarly, we observe that native IB and RDMAoE provide much better performance than TCP/IP and IPoIB. MPI running over RDMAoE has quite close latency to that of MPI running over native IB. The similar comparison is also seen in Figures 8 (a), (b) and (c) which show the latency results using multiple pairs of communications. However, here surprisingly, the performance of IPoIB for small messages becomes worse as compared to Figure 7 (a). We are investigating this currently.

2) Performance of Collective Operations: In this section, IMB [9] is utilized to measure the performance of MPI collectives over different communication protocols. Figures 9 and 10 present the results of *MPI_Allgather* and *MPI_Allreduce*, respectively. Except the variation in *MPI_Allgather* running with TCP/IP for medium messages, the comparison in all the cases are conformable to that for point-to-point communication. This means that RDMAoE can provide MPI with steady comparable performance to native IB not only in basic point-to-point communications but also in collective communication.

3) NAS Application Performance: We further utilize the NAS [3] benchmarks to compare the impact of different communication protocols. Due to large range of scales in different NAS applications, we normalized the execution time in all cases to the time in native IB case. As shown in

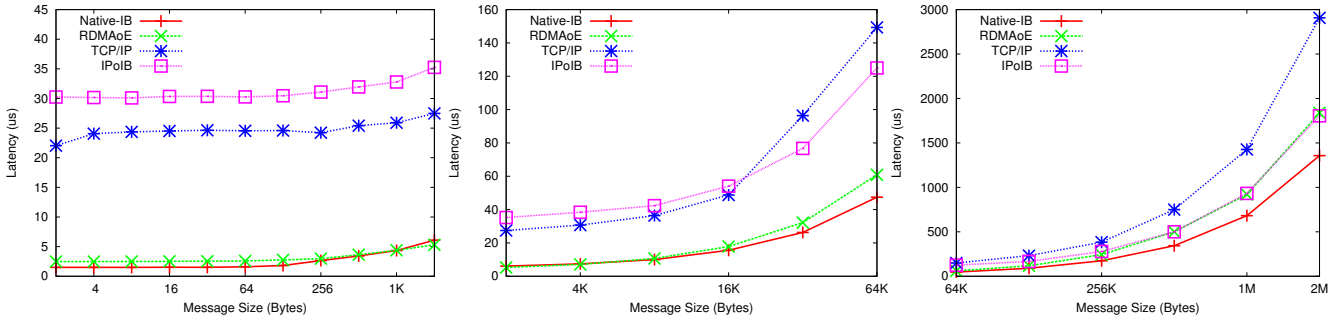

Figure 3. Back to Back Verbs Level Latency Comparison of: (a) Small messages, (b) Medium messages, and (c) Large messages

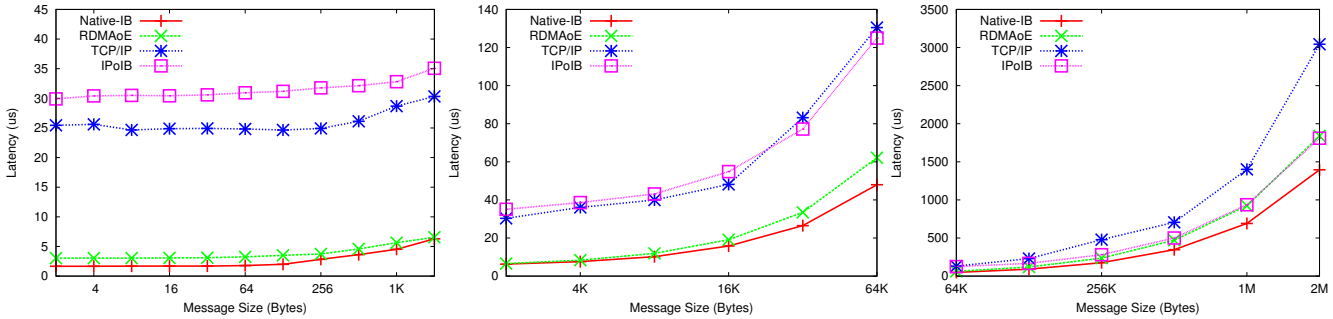

Figure 4. Verbs Level Latency Comparison of: (a) Small messages, (b) Medium messages, and (c) Large messages

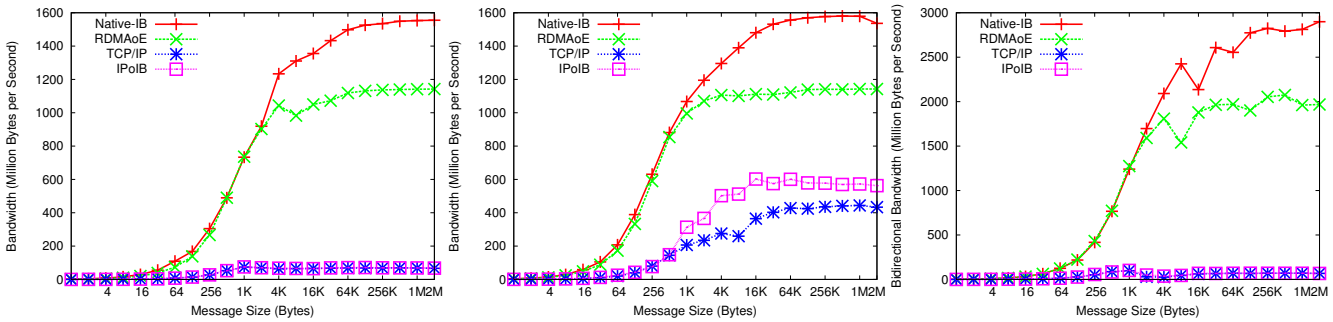

Figure 6. MPI Bandwidth Performance using OMB: (a) Uni-directional Bandwidth, (b) Uni-directional Multi-Pair Bandwidth and, (c) Bi-Directional Bandwidth

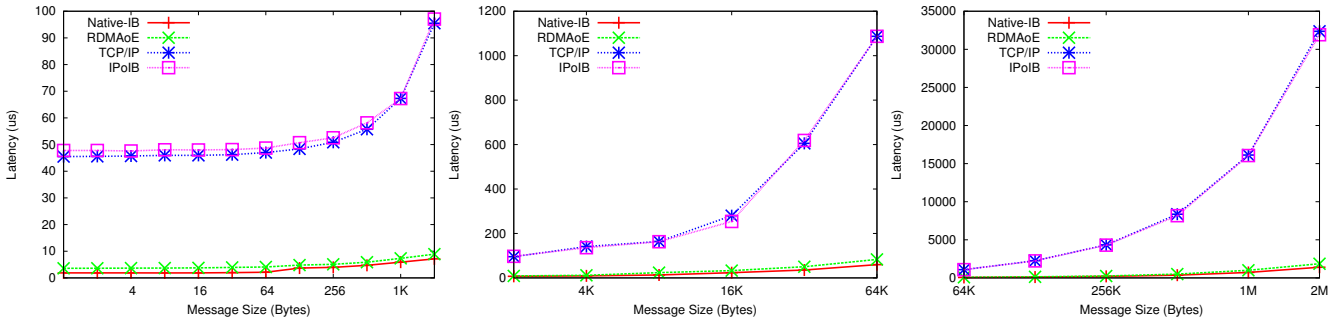

Figure 7. MPI Latency Performance using OMB: (a) Small messages, (b) Medium messages, and (c) Large messages

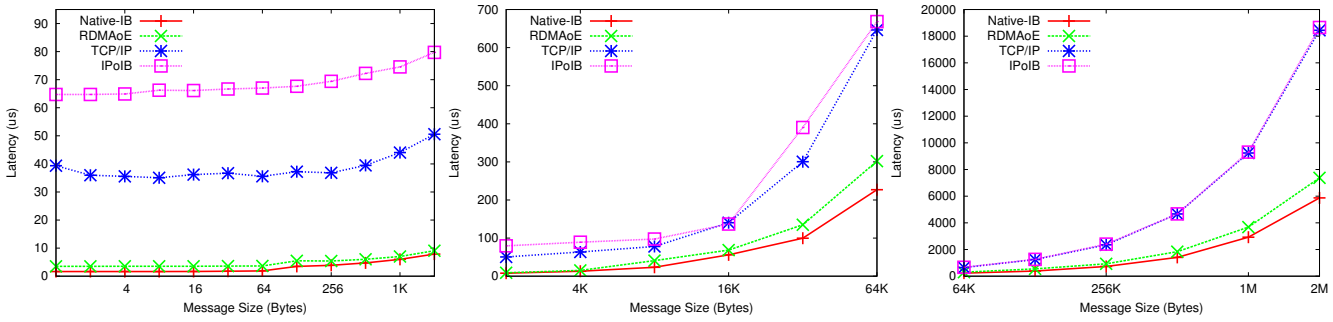

Figure 8. MPI Multi-Latency using OMB: (a) Small messages, (b) Medium messages, and (c) Large messages

Figure 11, native IB and RDMAoE provide much better performance in all the applications, about 3 to 4 times improvement on average. TCP/IP performs the worst and IPoIB performs a minor better than TCP/IP. These results are consistent with the previously seen results in Section III-B and Section III-C1.

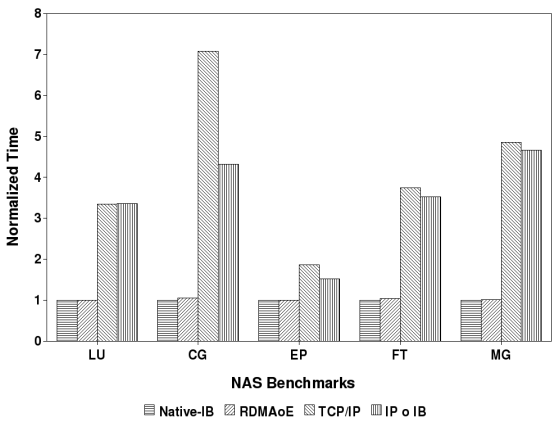

Figure 11. NAS Application Performance

D. Data-Center Application Performance

In this section, we use one popular application, i.e., FTP, used in many data centers to characterize the performance of the aforementioned communication protocols.

The existing FTP libraries such as GridFTP [2] are designed to execute on top of TCP, so they can not run over native IB or RDMAoE. In our previous work [10], we have designed and evaluated an IB based ADTS-FTP library, which can make use of the zero-copy benefits of native IB. Here, we further make some minor modifications (that is mentioned in Section II-B) to make it also run on RDMAoE. GridFTP is utilized to show the performance of TCP/IP and IPoIB. We also apply the well known tuning methods (such as adjusting the TCP buffer size etc.) to optimize its performance [10].

Figure 12 shows the file transferring time of *get* operation for varying file sizes. We observe that ADTS-FTP using native IB and RDMAoE performs much better than GridFTP over TCP/IP and IPoIB. This is resulted from the much lower overhead of zero-copy operations in the former case than the TCP related processing and copies in the latter case. RDMAoE can provide almost the same performance as the native IB for smaller file sizes and very small degradation as the file size increases. This still

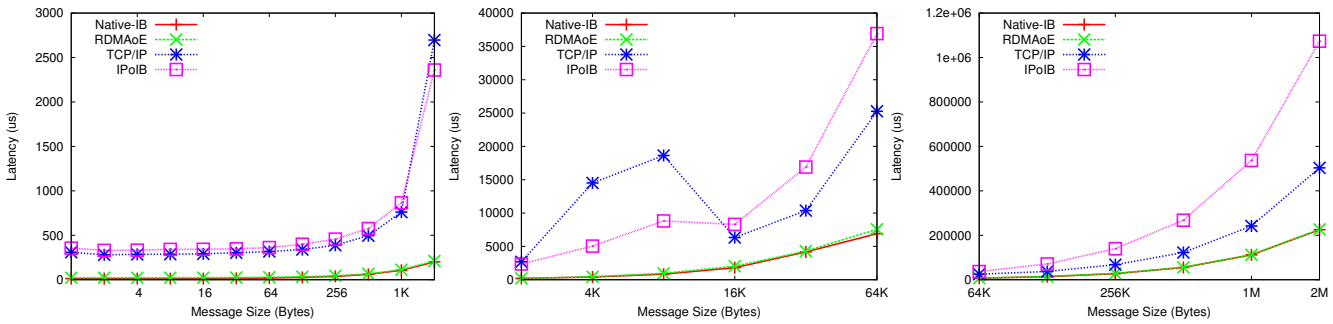

Figure 9. Performance Comparison Allgather Collective for: (a) Small messages, (b) Medium messages, and (c) Large messages

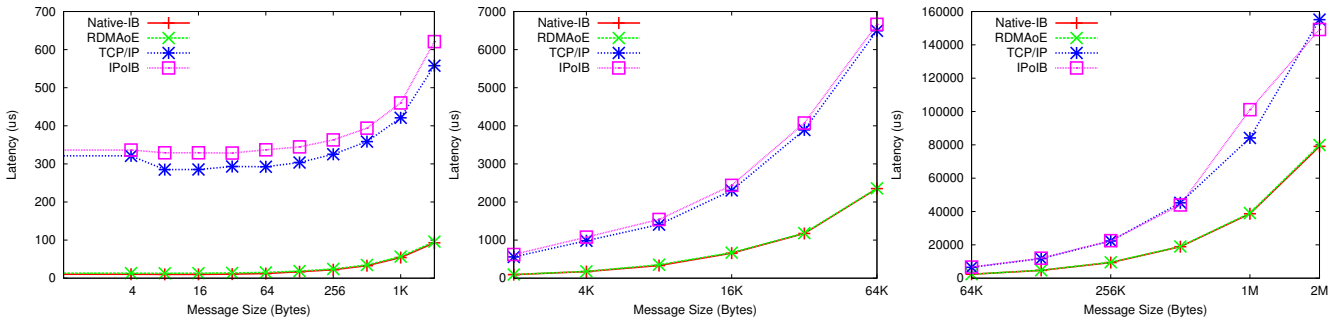

Figure 10. Performance Comparison Allreduce Collective for: (a) Small messages, (b) Medium messages, and (c) Large messages

tells us that RDMAoE is a good trade off between the performance benefits and the equipment cost.

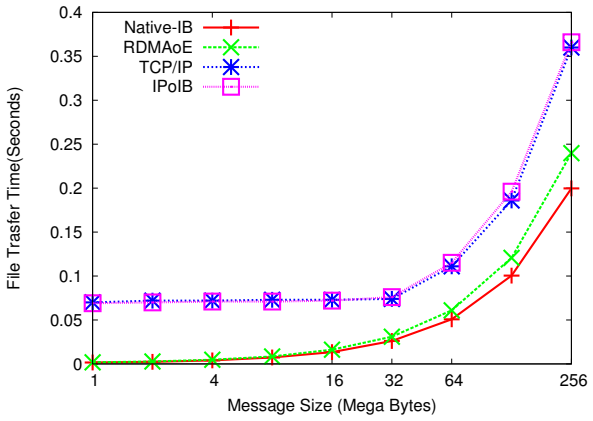

Figure 12. FTP *get* Operation Time

IV. Related Work

The current Ethernet is either slow with commodity NICs using generic TCP/IP stack or costly with TCP/IP Offload Engine (TOE). Many researchers have been paying effort to improve the situation.

Michael Oberg etc. [19] evaluated RDMA over Gigabit Ethernet as a potential Linux cluster interconnect. In the paper, they describe Ammasso Gigabit Ethernet RDMA technology, which uses a custom protocol to wrap RDMA method in TCP/IP packets and send them over Ethernet frames. The Ammasso hardware functions as a TCP offload engine, implementing a custom TCP/IP stack in hardware. Through both network level and application level experiments, they demonstrated better throughput and latency performance of Ammasso RDMA adapter than the typical non-RDMA Gigabit adapters in a commodity Ethernet environment.

Internet Wide Area RDMA Protocol (iWARP) standard [22], which was introduced to extend the benefits of RDMA to the traditional Ethernet-based networks, allows

for zero-copy transfer of data over the legacy TCP/IP communication stacks. In [18], Narravula et al. presented MPI-iWARP, a high performance MPI over iWARP as means to overcome the performance limitations observed in the traditional TCP/IP stack based software.

MX over Ethernet (MXoE) protocols [16] were proposed by Myricom in 2006, which support Myricoms dual-protocol Myri-10G network-interface cards and standard 10-Gigabit Ethernet switches to achieve latencies 5 to 10 times lower than with TCP/IP over Ethernet. MXoE extends advantages of MX [17] to standard 10-Gigabit Ethernet switching by kernel bypass, which can achieve low latency and low host-CPU utilization by allowing application programs to communicate directly with firmware in the programmable Myri-10G NICs. Open-MX [5] is a high-performance implementation of the Myrinet Express message-passing stack over generic Ethernet networks, which provides wire-protocol compatibility for applications with the native MXoE stack.

Other than the efforts that try to use zero-copy send/receive and RDMA technologies over Ethernet, there are also other works that have been done on different protocols.

The Joint Network Interface Controller (JNIC) project [23] developed a novel Ethernet endpoint to explore high-performance in-data-center communications over Ethernet. They designed and evaluated a network architecture for Ethernet-based communications in future data centers. The project demonstrated that NIC hardware closely coupled into the CPU/memory complex can be combined with on-load software running on a multi-core CPU to achieve increased Ethernet performance while reducing the endpoint costs.

V. Conclusion

We performed a comprehensive evaluation of four possible modes of communication that can be performed using current generation InfiniBand hardware at verbs, MPI, application and data center levels. The experimental results obtained from our evaluation shows that the new RDMAoE technology offers end users an easy and high performance solution to the problem of network convergence. From our verbs level latency tests, we can clearly see that RDMAoE is capable of providing latencies close to that provided by Native IB based applications over a standard 10 GigE link. As expected, the latency performance that IPoIB and standard TCP based solutions give is much lower than that given by Native IB and RDMAoE. The trends seen in the verbs level tests are seen to hold throughout the other evaluations done at a higher level such as micro-benchmark, collectives, application and data centers.

References

[1] IP over InfiniBand Working Group. `http://www.ietf.org/html.charters/ipoib-charter.html`.

[2] W Allcock. GridFTP: Protocol Extensions to FTP for the Grid. Global Grid ForumGFD-R-P.020,2003.

[3] D. H. Bailey, E. Barszcz, J. T. Barton, D. S. Browning, R. L. Carter, D. Dagum, R. A. Fatoohi, P. O. Frederickson, T. A. Lasinski, R. S. Schreiber, H. D. Simon, V. Venkatakrishnan, and S. K. Weeratunga. The NAS Parallel Benchmarks. volume 5, pages 63–73, Fall 1991.

[4] Elmar Bartel. http://hpux.connect.org.uk/hppd/hpux /Networking/Admin/nttcp-1.47/.

[5] Brice Goglin. Design and Implementation of Open-MX: High-Performance Message Passing over Generic Ethernet Hardware. In *CAC 2008: Workshop on Communication Architecture for Clusters, held in conjunction with IPDPS 2008*, Miami, FL, April 2008. IEEE Computer Society Press.

[6] IEEE 802.3 Ethernet Working Group. IEEE 802.3. http://www.ieee802.org/3/.

[7] IEEE802.3 10GBASE-CX4 Task Force. IEEE P802.3ak 10GBASE-CX4 Standard. `http://www.ieee802.org/3/ak/index.html/`.

[8] Infiniband Trade Association. http://www.infinibandta.org.

[9] Intel Corporation. Intel Micro Benchmarks. http://software.intel.com/en-us/articles/intel-mpi-benchmarks/.

[10] P. Lai, H. Subramoni, S. Narravula, A. Mamidala, and D. K. Panda. FTP Mechanisms for High Performance Data-Transfer over InfiniBand. In *The 38th International Conference on Parallel Processing (ICPP '09)*.

[11] Mellanox OFED Stack for Linux Users Manual. http://www.mellanox.com/pdf/products/software/Mellanox_OFED_Linux_User_Manual_1_20.pdf.

[12] Mellanox Technologies. ConnectX Architecture. http://www.mellanox.com/products/connectx_architecture.php.

[13] Mellanox Technologies. RDMA over Ethernet. `http://www.opensubscriber.com/message /general@lists.openfabrics.org/12419517.html/`.

[14] MPI Forum. MPI: A Message Passing Interface. In *Proceedings of Supercomputing*, 1993.

[15] MVAPICH2: High Performance MPI over InfiniBand and iWARP. http://mvapich.cse.ohio-state.edu/.

[16] Myricom. MX over Ethernet. http://www.myri.com/Myri-10G/documentation/Low_Latency_Ethernet.pdf.

[17] Myricom. Myrinet Express (MX): A High Performance, Low-Level, Message-Passing Interface for Myrinet. http://www.myricom/scs/MX/doc/mx.pdf.

[18] S. Narravula, A. Mamidala, A. Vishnu, G. Santhanaraman, and D. K. Panda. High Performance MPI over iWARP: Early Experiences. In *Int'l Conference on conference on Parallel Processing*.

[19] Michael Oberg, Henry M. Tufo, Theron Voran, and Matthew Woitaszek. Evaluation of RDMA Over Ethernet Technology for Building Cost Effective L inux Clusters. In *the 7th LCI International Conference on Linux Clusters*, May 2006.

[20] OSU Micro-benchmarks. http://mvapich.cse.ohio-state.edu/benchmarks/.

[21] RDMA Consortium. http://www.rdmaconsortium.org/home/draft-recio-iwarp-rdmap-v1.0.pdf.

[22] RDMAConsortium. Architectural Specications for RDMA over TCP/IP. http://www.rdmaconsortium.org/.

[23] Michael Schlansker, Nagabhushan Chitlur, Erwin Oertli, Paul M. Stillwell Jr, Linda Rankin, Dennis Bradford, Richard J. Carter, Jayaram Mudigonda, Nathan Binkert, and Norman P. Jouppi. High-performance Ethernet-based Communications for Future Multi-core Processors.