Performance Evaluation of RDMA over IP: A Case Study with Ammasso Gigabit Ethernet NIC

HYUN-WOOK JIN, SUNDEEP NARRAVULA, GREGORY BROWN, KARTHIKEYAN VAIDYANATHAN,
PAVAN BALAJI, AND DHABALESWAR K. PANDA

Technical Report
OSU-CISRC-6/05-TR40

Performance Evaluation of RDMA over IP: A Case Study with Ammasso Gigabit Ethernet NIC*

Hyun-Wook Jin Sundeep Narravula Gregory Brown Karthikeyan Vaidyanathan
Pavan Balaji Dhabaleswar K. Panda

Department of Computer Science and Engineering
The Ohio State University
2015 Neil Avenue
Columbus, OH 43210
{jinhy, narravul, browngre, vaidyana, balaji, panda}@cse.ohio-state.edu

Abstract

Remote Direct Memory Access (RDMA) has been proposed to overcome the limitations of traditional send/receive based communication protocols. The immense potential of RDMA to improve the communication performance while being extremely conservative on resource requirements have made RDMA the most sought after feature in current and next generation networks. Recently, there are many active efforts to enable RDMA over IP and the fabrication of RDMA-enabled Ethernet NICs has just been started. However, its performance has not been quantitatively evaluated over WAN environments while existing researches have been focused on LAN environments. In this paper, we evaluate the performance of RDMA over IP networks with Ammasso Gigabit Ethernet NIC while emulating high delay WANs and varying load on remote node. We observe that RDMA is beneficial especially under heavy load conditions. More importantly, even with a high delay, RDMA can provide better communication progress and requires less CPU resource as compared to the traditional sockets over TCP/IP. Further we show that RDMA can support high performance intra-cluster communications providing unified communication interface for inter- and intra-cluster communication. To the best of our knowledge, this is the first quantitative study of RDMA over IP on a WAN setup.

1 Introduction

Remote Direct Memory Access (RDMA) is emerging as the central feature in modern network interconnects. It has been proposed by the researchers to overcome the limitations of traditional communication protocols. RDMA protocols are extremely conservative on CPU and memory bandwidth usage. The immense potential of RDMA to improve the communication performance while being extremely conservative on resource utilization have made RDMA the most sought after feature in current and next generation networks. Interconnects like InfiniBand [3], Myrinet [5] and Quadrics [13] have long introduced RDMA in LAN environments.

RDMA over IP has been developed recently to extend the benefits of RDMA beyond the LAN environments and across the WAN/Internet. The RDMA consortium [6] has proposed the RDMA Protocol Verbs Specifications

*This research is supported in part by Department of Energy's Grant #DE-FC02-01ER25506, National Science Foundation's grants #CCR-0204429, #CCR-0311542, and CNS-0403342; and equipment donations from Ammasso, Inc. and Foundry Networks.

(RDMAVS 1.0) [8] to standardize the efforts. Several performance critical grid applications like GridFTP [14], GridRPC [11], etc. can benefit significantly by utilizing RDMA over IP. Further, web based applications like web-servers, data-centers, Internet proxies, etc. can also leverage the benefits of an IP based RDMA protocol for use over the WAN.

However, there have been no comprehensive quantitative evaluations of RDMA over WAN environments while some qualitative studies have been done [16, 7]. Although many researchers have shown the benefits of RDMA, these are limited to LAN environments [1, 12, 4]. There has been research on utilizing RDMA for IP network storage [10] but this does not take into account the high delay characteristics of WAN. Hence it has become very critical to evaluate and analyze the performance of RDMA over WAN environment.

The fabrication of RDMA-enabled Ethernet NICs has just been started. In this paper, we evaluate the performance of RDMA over WAN with Ammasso Gigabit Ethernet NIC [2] in several aspects of performance such as (i) basic communication latency, (ii) computation and communication overlap, (iii) communication progress, (iv) CPU resource requirements, and (v) unification of communication interface. These performance metrics and features are known to be important to user applications. We are focusing especially on the impact of larger delays often experienced for WAN communications upon these important performance factors. In order to emulate the WAN environment, we construct two different IP networks connected by a router. In addition, we implement a delay generator named *degen* on the router, which adds a delay to the network, characterizing WAN. Our experimental results clearly show that, even with a high delay, RDMA can provide better communication progress and less CPU resource requirements as compared to the traditional sockets over TCP/IP. We also observe that RDMA is beneficial especially under loaded conditions. Further we show that RDMA can support high performance intra-cluster communications providing unified communication interface for inter- and intra-cluster communication. To the best of our knowledge, this is the first quantitative study of RDMA over IP on a WAN setup.

The rest of the paper is organized as follows: In Section 2, we describe an overview of RDMA and Ammasso Gigabit Ethernet NIC. Section 3 details our experimental methodologies and performance evaluation results. Finally, we conclude the paper in Section 4.

2 Background

In this section, we briefly describe the necessary background information.

2.1 Remote Direct Memory Access

Remote Direct Memory Access (RDMA) is a modern network feature that allows nodes to communicate without any involvement of remote node's CPU. Two basic RDMA operations exist: (i) RDMA Write and (ii) RDMA Read. RDMA write is used to transfer data to a remote node's memory and RDMA Read is used to get data from a remote node. In a typical RDMA operation, the initiator node posts a descriptor containing the details of the data-transfer, which contains addresses of both local and remote buffers, to the NIC and the NIC handles the actual data-transfer asynchronously. On the remote side, the NIC stores/fetches the data to/from the host memory without disturbing the CPU. The usage of RDMA presents multi-fold benefits to enable application scalability. Since the sender and receiver CPU's are not involved in RDMA data-transfers, the applications benefit from this additional computing capability. Further, the elimination of kernel context switches and multiple copies during the data-transfers provides significantly better performance. Interconnects like InfiniBand [3], Myrinet [5] and Quadrics [13] have introduced RDMA in LAN environments. In addition, several protocols have been proposed to take advantage of RDMA operations on IP networks [15, 17, 9].

2.2 Overview of Ammasso Gigabit Ethernet NIC

The Ammasso Gigabit Ethernet Network Interface Card (NIC) [2] provides an implementation of the RDMA over TCP/IP enabled NIC. Based on the RDMA Protocol Verbs (RDMAVS 1.0) [8] specified by the RDMA consortium, the RDMA interface of the Ammasso Gigabit Ethernet NIC provides low latency and high bandwidth on Gigabit Ethernet network. As shown in Figure 1, Ammasso Gigabit Ethernet NIC supports the legacy sockets interface and the Cluster Core Interface Language (CCIL) interface. The CCIL interface is an implementation of the Verbs layer to utilize RDMA over IP. The CCIL interface uses the RDMA layer and offloaded TCP/IP on the NIC to transmit the data. On the other hand, the sockets interface still sends and receives the data through the traditional TCP/IP implemented in the operating system kernel. The CCIL interface enables zero-copy and kernel-bypass data transmission.

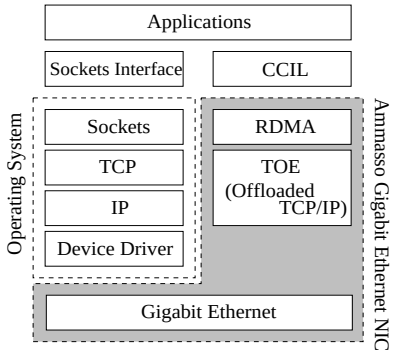


Figure 1. Protocol Stacks on Ammasso Gigabit Ethernet NIC

3 Evaluation of RDMA over IP Networks

In this section, we compare RDMA with traditional TCP/IP sockets on WAN environments with respect to (i) latency, (ii) computation and communication overlap, (iii) communication progress, (iv) CPU resource requirements, and (v) unification of communication interface.

3.1 Experimental WAN Setup

Our performance evaluations have been performed on the experimental system shown in Figure 2, where we have two different IP networks and they are connected through a workstation-based router. The end nodes are SuperMicro SUPER P4DL6 nodes of which each has dual Intel Xeon 2.4GHz processors with a 512KB L2 cache and an Ammasso 1100 Gigabit Ethernet NIC. The router node is a SuperMicro SUPER X5DL8-GG workstation with dual Intel Xeon 3.0GHz processors, 512KB L2 cache, and 2GB of main memory. The router node is connected to IP networks A and B with Broadcom BCM5703 and Intel PRO/1000 Gigabit Ethernet NICs, respectively. All nodes use Linux kernel version 2.4.20. The switches used for each IP network are Foundry FastIron Edge X448 Switch and Netgear GS524T Gigabit Switch, respectively.

To reflect the characteristics of high latency in the WAN environment, we have implemented a delay generator named *degen*. It delays the forwarding of packets on the router based on a given delay value. Each packet is time-stamped when it reaches the router. *Degen* uses this time stamp to delay the packet appropriately. We use the netfilter hooks provided by Linux 2.4 kernel to implement *degen*, which can be dynamically inserted to the chain of packet processing by using a run-time loadable kernel module. As shown in Figure 2, *degen* adds a delay to each packet after the routing decision has been taken place.

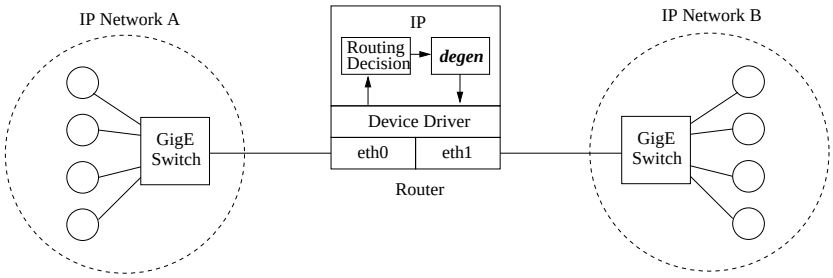


Figure 2. Experimental WAN Setup

3.2 Basic Communication Latency

In this section, we carry out the latency test in a standard ping-pong fashion to report one-way latency. Our results show, in the case that the application uses the same buffer for multiple communication, both RDMA and sockets shows almost the same latency. The reason is that although the sockets require the copy operation between user and kernel buffers, since the data can be in the cache the copy cost is not significant. On the other hand, if the application uses different buffers for every communication iterations (i.e., communication data is always out of cache) RDMA can achieve a better performance. For example, RDMA can achieve around 220μs less latency than the sockets with 32KB data. It is because RDMA performs the zero-copy data transmission and thus its performance is not varying between in-cache and out-of-cache data. However, we observed that such benefit is reducing as the network delay is increasing because the copy overhead is not the dominant overhead compared to the network delays larger than 1ms.

3.3 Computation and Communication Overlap

In this section, we evaluate how well the process is able to overlap computation with communication. In our test shown in Figure 3(a), the client performs a computation loop in between ping (i.e., sending) and pong (i.e., receiving) operations of the latency test described in Section 3.2. We evaluate the computation and communication overlap ratio with $(Computation\_Time)/(Total\_Time)$ on the client side. Thus a value closer to 1 represents a better computation and communication overlap. Figure 4(a) shows the overlap ratios varying computation time without network delay. The message size used is 1KB. As we can see, RDMA can achieve better overlap even with smaller computation time compared to the sockets. This is because RDMA provides asynchronous communication interface. Second, we do not need to pay CPU resources to get the data from remote node because the remote node uses RDMA write. Further, the offloaded TCP/IP increases the chance of overlapping between packet processing overhead and computation overhead. Figure 4(b) shows the overlap ratio values of RDMA and sockets for varying network delay. It can be observed that the values are almost the same with a large network delay. It is mainly because the network delay is the critical overhead and it can be overlapped with computation time regardless of RDMA or sockets. Since the packet processing overhead of end nodes is not a critical overhead anymore on WAN, its overlapping with other overheads does not affect much to the overlap ratio. However, still we can see that RDMA can provide better overlap than sockets for delays in order of a few milliseconds. It is to be noted that our sockets latency test is using non-blocking sockets to maximize the computation and communication overlapping.

3.4 Communication Progress

In many distributed systems, we often observe the communication pattern that a node requests some data to a remote node and it returns the data, which can be implemented with either pair of send and receive or RDMA

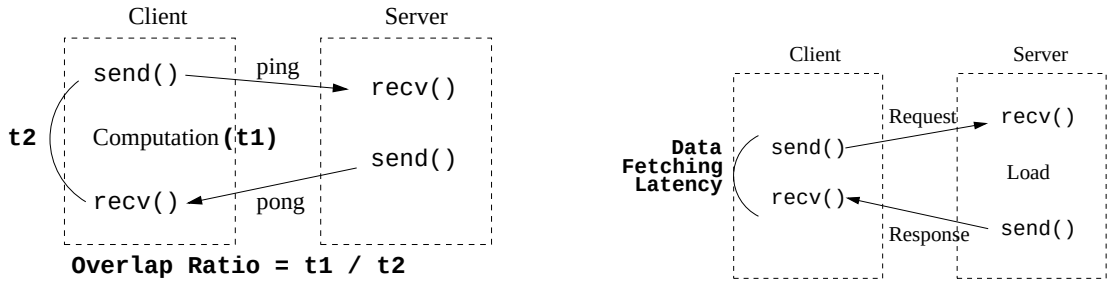


Figure 3. Pseudo Code for Benchmarks: (a) Overlap Test and (b) Communication Progress Test

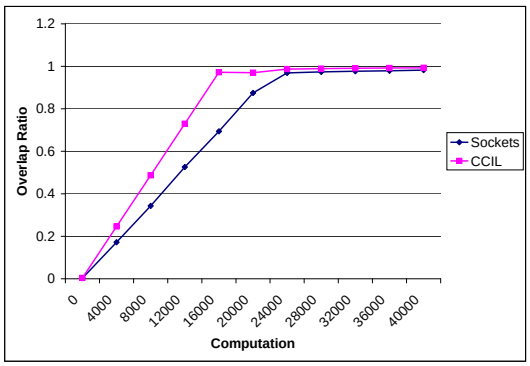
read. Moreover, the remote node can be heavily loaded because of burst requests on the data or CPU intensive computations. To compare RDMA read with traditional sockets in this scenario, we simulate a load on the remote node by adding a response delay to the remote process and measure the latency to get the data from remote node as shown in Figure 3(b). Figure 5 shows the latency results varying the load on the remote node and the network delay, respectively. Since RDMA read does not require any involvement of remote process for data transmission, it can read data from remote memory without any impact from the load on the target. It is to be noted that the sockets interface is not able to deliver good performance as the load increases. With increase in network delay, both the interfaces perform similarly because the network delay tends to dominate the performance costs.

3.5 CPU Resource Requirements

To see how much CPU resource requirements for communication affect the application performance, we run an application that performs basic mathematical computations while 10 clients keep on sending data of 64KB to this node. We report the total execution time of the application under this scenario. Figure 6 shows that the sockets interface significantly degrades the application performance even on high delay WANs. The reason is that, in the case of sockets, the remote node has to involve in the communication with respect to interrupts and packets processing at the kernel level and posting receive requests at the application level. This results in stealing of the CPU resource from the computing application. However, RDMA can place the data to the remote memory without any CPU requirement on the remote node. Hence we can see in the figure that the application execution time is constant for all network delays.

3.6 Unification of Communication Interface

In addition to direct performance metrics detailed in the previous sections, WAN and LAN interoperability is a very important feature of RDMA over IP. Scenarios in which multiple inter-cluster and intra-cluster nodes communicate with each other need common communication interfaces for the job. Traditionally, sockets over TCP/IP has been the main protocol with this feature. However, with RDMA over IP, this interoperability can be achieved and it can be achieved with all the benefits described in the previous sections. Further, RDMA over IP performs significantly better then sockets for within LAN communications. Figure 7 shows the latency of CCIL and Sockets communications within a LAN. We see that the small message latency differs by almost 50% with RDMA being better. Hence, RDMA over IP benefits these multi-cluster applications with better communication both over the WAN as well as in the LAN.



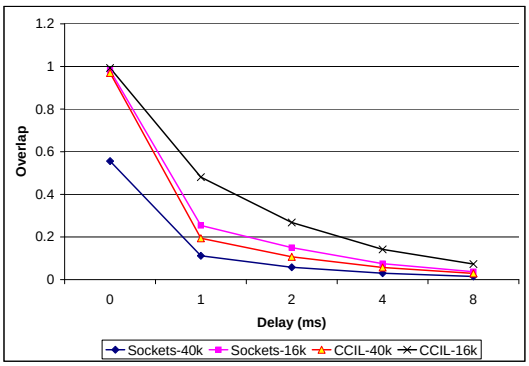
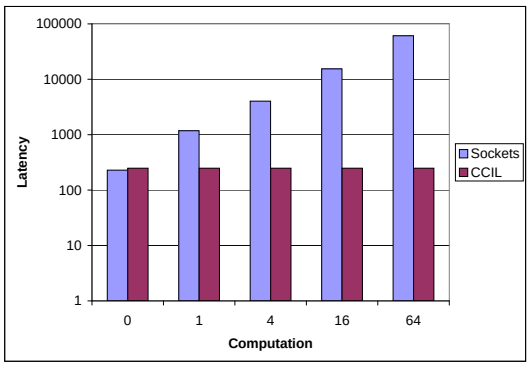


Figure 4. Overlap Ratio: (a) Varying Computation Time and (b) Varying Network Delay



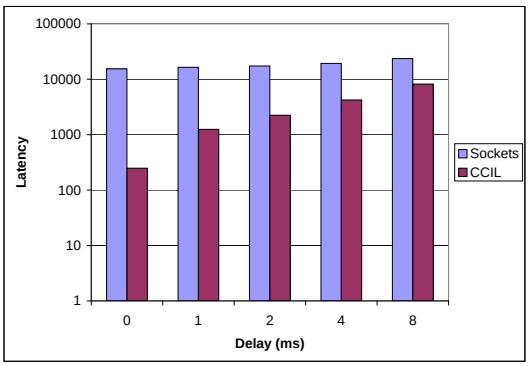


Figure 5. Data Fetching Latency: (a) Varying Load and (b) Varying Network Delay

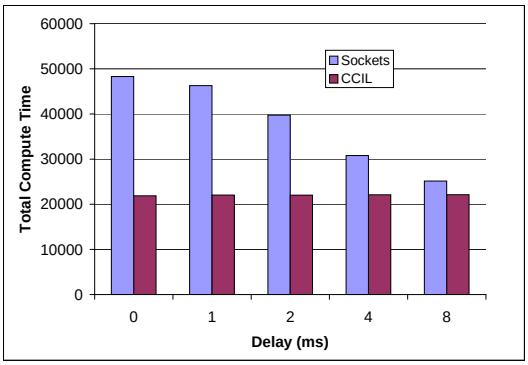


Figure 6. Impact on Application Execution Time on Different Network Delay

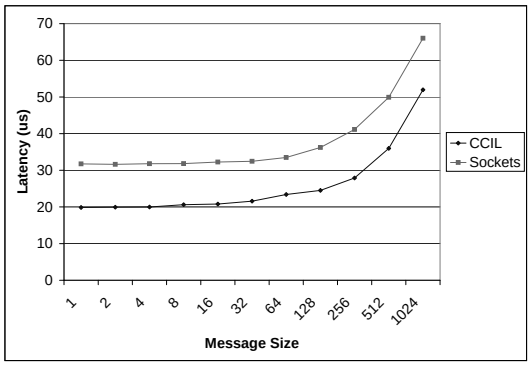


Figure 7. Communication Latency within a LAN with varying Message Sizes

4 Concluding Remarks

To resolve the limitations of existing communication semantics of send and receive, RDMA has been proposed in modern high-speed LAN interconnects. Moreover, recently there are many active efforts to enable RDMA over IP networks. However, its performance has not been quantitatively evaluated over WAN environments.

In this paper, we evaluate the performance of RDMA over WAN environments with Ammasso Gigabit Ethernet NIC. To reflect the characteristics of WAN environments we emulate it with a router with varying the high network delay. Our experimental results have revealed that, even with a high delay, RDMA can provide better communication progress and save CPU resource as compared to the traditional sockets over TCP/IP. This is mainly due to the fact that RDMA does not require involvement of remote side and the Offloaded TCP/IP leverages the benefit of RDMA over IP. As a result, we have presented the potential benefits of RDMA over IP networks with comprehensive performance evaluation.

We intend to continue working in this direction. We plan to improve the delay generator (*degen*) to reflect more characteristics of WANs and evaluate the performance of RDMA over IP with more applications. We also plan to design and evaluate RDMA-aware middleware for widely distributed systems over WAN.

References

[1] MPI over InfiniBand Project. http://nowlab.cis.ohio-state.edu/projects/mpi-iba/.

[2] Inc. Ammasso. http://www.ammasso.com.

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

[4] P. Balaji, K. Vaidyanathan, S. Narravula, K. Savitha, H. W. Jin, and D. K. Panda. Exploiting Remote Memory Operations to Design Efficient Reconfiguration for Shared Data-Centers. In *Workshop on Remote Direct Memory Access (RDMA): Applications, Implementations, and Technologies (RAIT)*, San Diego, CA, Sep 20 2004.

[5] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W. K. Su. Myrinet: A Gigabit-per-Second Local Area Network. http://www.myricom.com.

[6] RDMA Consortium. Architectural Specifications for RDMA over TCP/IP. `http://www.rdmaconsortium.org/`.

[7] W. Feng, G. Hurwitz, H. Newman, S. Ravot, L. Cottrell, O. Martin, F. Coccetti, C. Jin, D. Wei, and S. Low. Optimizing 10-Gigabit Ethernet for Networks of Workstations, Clusters and Grids: A Case Study. In *SC2003: High-Performance Networking and Computing Conference)*, November 2003.

[8] J. Hilland, P. Culley, J. Pinkerton, and R. Recio. RDMA Protocol Verbs Specification (Version 1.0). Technical report, RDMA Consortium, April 2003.

[9] M. Ko, J. Hufferd, M. Chadalapaka, Uri Elzur, H. Shah, and P. Thaler. iSCSI Extensions for RDMA Specification (Version 1.0). Technical report, RDMA Consortium, July 2003.

[10] K. Magoutis, S. Addetia, A. Fedorova, and M. Seltzer. Making the Most out of Direct Access Network-Attached Storage. In *Second USENIX Conference on File and Storage Technologies (FAST'03)*, March 2003.

[11] H. Nakada, S. Matsuoka, K. Seymour, J. Dongarra, C. Lee, and H. Casanova. A GridRPC Model and API. Technical report, GridRPC Working Group, November 2003.

[12] S. Narravula, P. Balaji, K. Vaidyanathan, S. Krishnamoorthy, J. Wu, and D. K. Panda. Supporting Strong Coherency for Active Caches in Multi-Tier Data-Centers over InfiniBand. In *SAN*, 2004.

[13] F. Petrini, W. C. Feng, A. Hoisie, S. Coll, and E. Frachtenberg. The Quadrics Network (QsNet): High-Performance Clustering Technology. In *Hot Interconnects*, 2001.

[14] Globus Project. GridFTP: Universal Data Transfer for the Grid. Technical report, The University of Chicago and The University of Southern California, September 2000.

[15] R. Recio, P. Culley, D. Garcia, and J. Hilland. An RDMA Protocol Specification (Version 1.0). Technical report, RDMA Consortium, October 2002.

[16] A. Romanow. An Overview of RDMA over IP. In *First International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2003)*, February 2003.

[17] Hemal V. Shah, James Pinkerton, Renato Recio, and Paul Culley. Direct Data Placement over Reliable Transports, November 2002.