Performance of HPC Middleware over InfiniBand WAN

SUNDEEP NARRAVULA, HARI SUBRAMONI, PING LAI,
BHARGAVI RAJARAMAN, RANJIT NORONHA, DHABALESWAR K. PANDA

Technical Report
OSU-CISRC-12/07-TR77.

Performance of HPC Middleware over InfiniBand WAN [*]

S. Narravula H. Subramoni P. Lai B. Rajaraman R. Noronha D. K. Panda

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

Abstract

High performance interconnects such as InfiniBand (IB) have enabled large scale deployments of High Performance Computing (HPC) systems. High performance communication and IO middleware such as MPI and NFS over RDMA have also been redesigned to leverage the performance of these modern interconnects. With the advent of long haul InfiniBand (IB WAN), IB applications now have inter-cluster reaches. While this technology is intended to enable high performance network connectivity across WAN links, it is important to study and characterize the actual performance that the existing IB middleware achieve in these emerging IB WAN scenarios.

In this paper, we study and analyze the performance characteristics of the following three HPC middleware: (i) IPoIB (IP traffic over IB), (ii) MPI and (iii) NFS over RDMA. We utilize the Obsidian IB WAN routers for inter-cluster connectivity. Our results show that many of the applications absorb smaller network delays fairly well. However, most approaches get severely impacted in high delay scenarios. Further, communication protocols need to be optimized in higher delay scenarios to improve the performance. In this paper, we propose several such optimizations to improve communication performance. Our experimental results show that techniques such as WAN-aware protocols, transferring data using large messages (message coalescing) and using parallel data streams can improve the communication performance (upto 50%) in high delay scenarios. Overall, these results demonstrate that IB WAN technologies can enable cluster-of-clusters architecture as a feasible platform for HPC systems.

Keywords: *Cluster-of-Clusters, InfiniBand, MPI, MVAPICH2, IPoIB, NFS, Obsidian Longbow XR, InfiniBand WAN*

1. Introduction

Ever increasing demands for High Performance Computing (HPC) systems and high performance to cost ratios have led to the growth and popularity of commodity clusters. Modern interconnects like InfiniBand have immensely enhanced the performance achieved by such clusters.

Further, organizations often need to deploy newer clusters to accommodate their increasing compute demands. The multi-cluster scenarios in which these deployments are made usually vary, with the clusters being within the same room, building, or campus or across multiple geographically separated campuses. Such deployment scenarios are usually driven by administrative and engineering considerations like power and cooling restrictions, space constraints, etc. Due to these emerging trends, organizations often find themselves with multiple fragmented clusters forming cluster-of-clusters. Figure 1 shows a typical cluster-of-clusters scenario.

While these clusters are often equipped with high performance modern interconnects for intra-cluster communication, they usually depend on TCP/IP for their inter-cluster communications requirements. This is largely due to the fact that InfiniBand fabrics have typically been limited to cable lengths of up to 20 meters. While these cable lengths are acceptable (with some constraints for very large clusters) to a certain extent, they fail to extend the reach of InfiniBand fabrics beyond a single machine room or a building. This imposes a severe performance penalty on utilizing cluster-of-clusters for HPC.

To address this problem, recently, InfiniBand range extenders like Intel Connects [5] and Obsidian Longbows [8] have been introduced. Intel Connects can extend the reach of IB fabrics upto 100 meters and the Obsidian Longbows are capable of covering Wide Area Network (WAN) distances. While this IB WAN technology provides essential capabilities for IB range extensions, it is also very important to evaluate and understand these capabilities and

[*] This research is supported in part by DOE grants #DE-FC02-06ER25749 and #DE-FC02-06ER25755; NSF grants #CNS-0403342 and #CCF-0702675; grants from Intel, Mellanox, Cisco systems, Linux Networx and Sun Microsystems; and equipment donations from Intel, Mellanox, AMD, Apple, Appro, Chelsio, Dell, Fujitsu, Fulcrum, Microway, Obsidian, PathScale, IBM, SilverStorm and Sun Microsystems.

the limitations thereof.

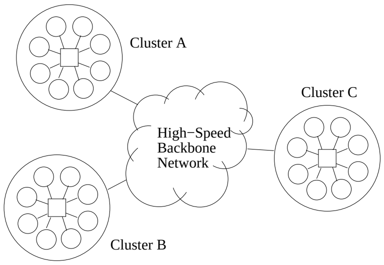

Figure 1. A Cluster-of-Clusters Scenario

On the other hand, existing IB applications and widely used libraries such as MPI [18], NFS over RDMA, etc. are usually developed based on the assumptions about IB fabrics which hold true in intra-cluster environments. However, in WAN scenarios these assumptions might not hold and can lead to significant performance degradation. In particular, a latency addition of about 5 us per km of distance is observed and these larger wire latencies cannot be hidden from the applications. As an example of such a protocol we present the following: an optimization that several MPI libraries use the rendezvous protocol [14] for medium and large message transfers. These protocols rely on the trade-offs between multiple message copies and rendezvous message exchanges. The costs of such protocols change significantly over WAN communication links. Further, the WAN separations often vary and can be dynamic in nature. Hence, the communication protocols used for IB WAN need to be re-designed.

In this context, several researchers [6, 11, 19] have looked at basic performance evaluations of certain applications and middleware. However, it is important to perform a detailed study of the performance characteristics of HPC middleware and applications in varying cluster-of-clusters scenarios. i.e. a thorough understanding of IB WAN communications is needed for different transport protocols with respect to WAN delays and communication patterns in order to effectively redesign existing HPC middleware and design the next generation's HPC systems.

In this paper, we take on these challenges and carry out in-depth performance study of various HPC middleware with IB WAN, carry out sensitivity study with varying WAN delays, re-design internal protocols of the middleware and evaluate the performance of the new designs. In particular, the following are our main contributions:

- Study and analyze the general communication performance of HPC middleware, including (i) IPoIB, (ii) MPI and (iii) NFS over RDMA, in different cluster-of-clusters scenarios
- Propose basic design optimizations for enhancing communication performance over WAN
- Internal protocols of the middleware are enhanced to demonstrate the potential benefits thereof
- Study the overall feasibility of cluster-of-clusters architecture as a platform for HPC systems

Our experimental results show that all communication protocols can absorb small WAN (upto 100us) delays and sustain performance. Also, as can be expected, utilizing large message transfers and parallel communication streams improves the bandwidth utilization of the WAN link significantly. We observe an improvement of upto 50% for parallel stream communication and an improvement of upto 90% for hierarchical collectives in high delay networks. Further, by tuning protocols in existing middleware like MPI, we see an improvement of up to 83% in certain cases for basic message passing. We also observe that applications like IS and FT show considerable tolerance to the higher latencies seen in WAN environments. Overall, our results demonstrate the feasibility of Obsidian Longbow IB range extenders to create high performance cluster-of-clusters architectures.

The remainder of this paper is organized as follows: Section 2 gives a brief overview of InfiniBand, InfiniBand WAN, NFS over RDMA and MPI. In Section 3 we present the detailed microbenchmark level evaluations of Obsidian Longbows in different cluster-of-clusters scenarios. We further analyze the performance of IPoIB, MPI and NFS over RDMA in these cluster-of-clusters scenarios in this section. Section 4 describes the related work. Finally, we summarize our conclusions and possible future work in Section 5.

2. Background

In this section we present a brief overview of InfiniBand, InfiniBand WAN, MPI over InfiniBand and NFS over RDMA.

2.1. InfiniBand

InfiniBand Architecture (IBA) [4] is an industry standard that defines a System Area Network (SAN) to design clusters offering low latency and high bandwidth. Increasing number of InfiniBand clusters are currently being deployed in several HPC scenarios including high performance computing systems, web and Internet data-centers, etc. IBA supports two types of communication semantics: Channel Semantics (Send-Receive communication model) and Memory Semantics (RDMA communication model). Remote Direct Memory Access (RDMA) [7] operations (including RDMA read and write) allow processes to access (read or write) the memory of a process on a remote node without the remote node's CPU intervention.

InfiniBand supports multiple transport mechanisms. *Reliable Connected* (RC) transport provides a connected

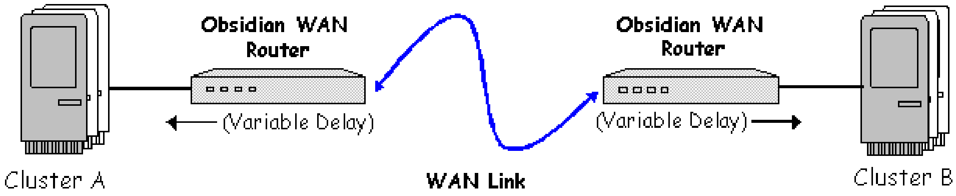

Figure 2. Cluster-of-Clusters Connected with Obsidian Longbow XRs

mode of transport with complete reliability. It supports communication using both channel and memory semantics and can transfer messages of sizes up to 4GB. On the other hand, *Unreliable Datagram* (UD) is a basic transport mechanism that can communicate over unconnected modes without reliability and can only send messages of up to the IB MTU size only. Further, this mode of communication does not support RDMA operations.

Although native IB protocol provides superior performance, most legacy applications and middleware are still based on TCP protocol. IPoIB driver [2] enables IP traffic over IB fabric and is one of the most popular protocols used in the IB networks. Currently, IB software stack supports both RC and UD based implementations of IPoIB.

2.1.1 InfiniBand Range Extension

Obsidian Longbows [8] primarily provide range extension for InfiniBand fabrics over modern 10 Gigabits/s Wide Area Networks (WAN). The Obsidian Longbows work in pairs establishing point-to-point links between clusters with one Longbow at each end of the link. Figure 2 shows a typical deployment of the IB WAN routers. The Longbows communicate using IPv6 Packets over SONET, ATM, 10 Gigabit Ethernet and dark fiber applications. The Longbows can essentially support IB traffic at SDR rates (8 Gbps). To make up for the remaining 2 Gbps bandwidth, these Obsidian Longbow routers can also encapsulate a pair of 1 Gigabit/s Ethernet traffic across the WAN link.

In the basic switch mode, the Longbows appear as a pair of two-ported switches to the InfiniBand subnet manager as shown in Figure 2. Both the networks are then unified into one InfiniBand subnet which is transparent to the InfiniBand applications and libraries, except for the increased latency added by the wire delays.

The Obsidian Longbow XR routers also provide a highly useful feature of adding delay to packets transmitted over the WAN link. Each of the Longbows provide a web interface to specify delay. The packets are then delayed for the specified time before and after traversing over the WAN link. This added delay can indirectly be used as a measure of emulated distance. i.e. this essentially corresponds to the wire delay of about 5 *us* for each *km* of wire length. We leverage this feature to emulate cluster-of-clusters with varying degrees of separation in the following experiments.

2.2. MPI over InfiniBand

Message Passing Interface (MPI) [12] is one of the most popular programming models for writing parallel applications in cluster computing area. MPI libraries provide basic communication support for a parallel computing job. In particular, several convenient point to point and collective communication operations are provided. High performance MPI implementations are closely tied to the underlying network dynamics and try to leverage the best communication performance on the given interconnect. In this paper we utilize MVAPICH2 [18] for our evaluations. However, our observations in this context are quite general and they should be applicable to other high performance MPI libraries as well.

2.3. NFS over RDMA

NFS [3] has become the *de facto* standard for file-sharing in a distributed environment. It is based on single-server multiple-clients model, and communication between the server and the client is via Open Network Computing (ONC) remote procedure call (RPC). Traditionally, TCP or UDP is used as the underlying transport protocol. However, the performance and scalability is limited due to the overhead from the two-sided operations of these protocols.

Recently, with the emergence of high performance interconnects such as InfiniBand, NFS has been redesigned by utilizing the advanced features provided, e.g., RDMA mechanism is proposed as an alternative transport to reduce the copy overhead and CPU utilization. Researchers in [17] proposed a NFS/RDMA design in which the NFS server uses RDMA operations to perform the data transfers required for the NFS operations and showed that this approach shows significantly better scalability and performance as compared to the NFS over TCP or UDP for intra cluster scenarios.

3. Experimental Evaluation

In this section we present our evaluation methodology followed by detailed performance evaluations of basic IB communication over WAN and IB communication/IO middleware (including IPoIB, MPI and NFS over RDMA) using the Obsidian Longbow routers. To evaluate these components, we emulate different cluster-of-clusters scenarios with varying degrees of separation (wire length between clusters) by adding network delay at the Obsidian routers. Each microsecond of emulated delay corresponds to about 200m of wire length.

3.1. Methodology

In order to study and analyze the performance of IB communication and IO middleware, we first perform a basic low-level evaluation of IB protocols. These results provide a base line for understanding the results for higher level protocols. We perform all the tests with varying WAN delays. This corresponds to a cluster seperation of one Km for 5 us as shown in Table 1. We then evaluate and examine the performances of IPoIB (with both RC and UD transports), MPI and NFS (with RDMA and IPoIB). For all these scenarios, we perform basic tests followed by optimized tests such as parallel stream tests. Further, in order to examine the effect of WAN delay on applications and to study the overall utility of IB WAN for cluster of cluster scenarios, we utilize NAS benchmarks with MPI running over IB WAN.

Experimental Testbed: In our experiments we use the following two clusters connected by a pair of Obsidian Longbow XRs: (i) *Cluster A* consists of 32 Intel Xeon dual 3.6 Ghz processor nodes with 2GB of RAM and (ii) *Cluster B* consists of 64 Intel Xeon Quad dual-core processor nodes with 6GB RAM. Both the clusters are equipped with IB DDR memfree MT25208 HCAs and OFED 1.2 [10] drivers were used. The OS used was RHEL4U4. The WAN experiments are executed using nodes from each of the clusters as shown in Figure 2.

3.2. Basic 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 in cluster-of-clusters scenarios. The experiments evaluate the latency, bandwidth and bidirectional bandwidth between the nodes of the two clusters shown in Figure 2.

Table 1. Delay Overhead corresponding to Wire Length

Distance	Delay
1 (km)	5 (us)
2 (km)	10 (us)
20 (km)	100 (us)
200 (km)	1000 (us)
2000 (km)	10000 (us)

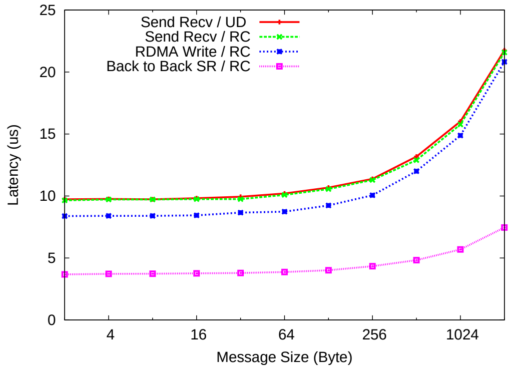

Figure 3. Verbs-level Latency

3.2.1 Verbs-level Latency

The increased latency due to longer range IB communications cannot be hidden from the IB applications. In the cluster-of-cluster situation, the total communication latency for small messages across a pair of Longbows is roughly the sum of the basic IB point-to-point latency, the latency added due to the two Longbows and the wire latency. The first two components are usually constant for small messages and the third component is dependent on the length of the communication path. This wire latency also becomes the dominant cost in small message latency in case of longer network separations as shown in Table 1.

In order to evaluate the first two components, we minimize the wire length (i.e., latency is 0 us) and measure the latency observed with and without the Longbow routers. Figure 3 shows the latency measured for different communication semantics and IB transport protocols. We observe that the pair of Longbows adds a latency of about 5 us as compared to the basic IB latency with back-to-back connected nodes, and the RDMA operations still outperforms the send/receive operation in this cluster-of-cluster configuration. (It is to be noted that both the clusters are DDR capable and hence the back-to-back latency is quite low).

3.2.2 Verbs-level Bandwidth

The Obsidian Longbows are capable of providing full bandwidth at SDR rates. We measure the bandwidth performance across our clusters (with increasing network delays) using RC and UD transports, respectively.

Verbs-level UD Bandwidth: In this experiment, we utilize *perftests* to measure the Send/Recv UD bandwidth with varying network delays. We observe that the bandwidth seen in this context is independent of the network delay. We achieve a peak bandwidth of about 967 MillionBytes/sec for a message size of 2k in all cases. This is primarily due to the fact that UD bandwidth tests do not involve any acknowledgements from the remote side and the data can be pushed at the full rate possible. Figure 4(a) which shows the UD bandwidth performance, indicates that UD is scalable with higher delays. It is to be noted that higher level protocols using UD transport will need to include their own reliability/flow control mechanisms (such as message acks, etc.) which can impact the performance.

We observe similar trends in Figure 4(b) for bidirectional bandwidth as well which shows a peak of about 1949 MillionBytes/sec.

Verbs-level RC Bandwidth: Figure 5(a) shows the bandwidth using RC transport mode, with varying delay between the clusters. We observe a peak bandwidth of about 984 MillionBytes/sec in all cases. However, the bandwidth observed for small and medium messages is progressively worse with increasing network delays. i.e. in order to leverage the high bandwidth capability of the IB WAN connectivity under higher network delays, larger messages need to be used. This is due to the fact that RC guarantees reliable and in-order delivery by ACKs and NACKs. This limits the number of messages that can be in flight to a maximum supported window size. While using larger messages, the pipeline can be filled with fewer messages, so it is seen that larger messages do quite well with larger delays. Higher level applications can fill the message transmission pipelines well in several different ways including message coalescing, overlapping multiple streams, etc.

Figure 5(b) shows the bidirectional bandwidth between the clusters under varying network delays. We observe trends similar to those seen for bandwidth in these as well. The peak bidirectional bandwidth seen is about 1960 MillionBytes/sec.

3.3. Performance of TCP/IPoIB

In this section, we aim to characterize the IPoIB throughput and provide insights to the middleware and application design in the cluster-of-clusters scenarios. Four main factors affect the bandwidth performance, i.e., MTU size, the TCP buffer size, the number of parallel streams and the WAN delays. Therefore, we vary these parameters in the following experiments. Messages with size 2M are used in all the experiments.

IPoIB UD Bandwidth: We evaluate the IPoIB bandwidth using the UD transport with varying WAN delays in both the single-stream and the parallel streams tests. Also, we vary the protocol window sizes in the single-stream experiment and the number of connections in the parallel stream experiment. The results are shown in Figures 6 (a) and (b), respectively. The MTU size used for IPoIB UD is 2KB.

From Figure 6(a), we see that larger bandwidth is achieved with larger window sizes. It is well known that TCP needs larger window sizes in order to achieve good bandwidth over large bandwidth networks. However, when the WAN delay increases, we observe that the performance of all the cases degrades. It is to be noted that the peak bandwidth that IPoIB UD achieves is significantly lower than the peak verbs-level UD bandwidth due to the TCP stack processing overhead. Overall, the default window size (>1M) in Figure 6(a) shows good performance in most cases. Thus, we use this default window size in all of the following experiments.

In order to improve the overall bandwidth performance, we measure the parallel stream bandwidth with various WAN delays as shown in Figure 6(b). We see that by using more streams, significant improvements (of up to 50%) are achieved in the higher delay scenarios. We observe that the peak IPoIB-UD bandwidth can be sustained even with the delay of 1ms using multiple streams. This is because of the fact that higher number of TCP streams lead to more UD packets with independent flow control (at TCP level), allowing for better utilization of the IB WAN long haul pipe, i.e. there are more outstanding packets that can be pushed out from the source at any given time frame.

IPoIB RC Bandwidth: For the IPoIB using RC transport mode, we also evaluate the single-stream and the parallel stream bandwidth with various WAN delays. One significant advantage of using RC transport mode for IPoIB is the that RC can handle larger packet sizes. This has the following advantages: (i) larger packets can achieve better bandwidth and (ii) per byte TCP stack processing decreases.

As expected in Figure 7(a), we see that the best bandwidth of 890 MillionBytes/sec is achieved with largest MTU size of 64KB (the maximum allowed for an IP packet). This is significantly higher than the bandwidth achieved for IPoIB-UD. That is because the IPoIB-UD test has an MTU size of just 2KB, which means that more packets need to be transferred for the same amount of data and correspondingly more overhead is introduced. In addition, the number of packets required to utilize the WAN link bandwidth fully is significantly higher. On the other hand, we also observe that the bandwidth drops sharply with

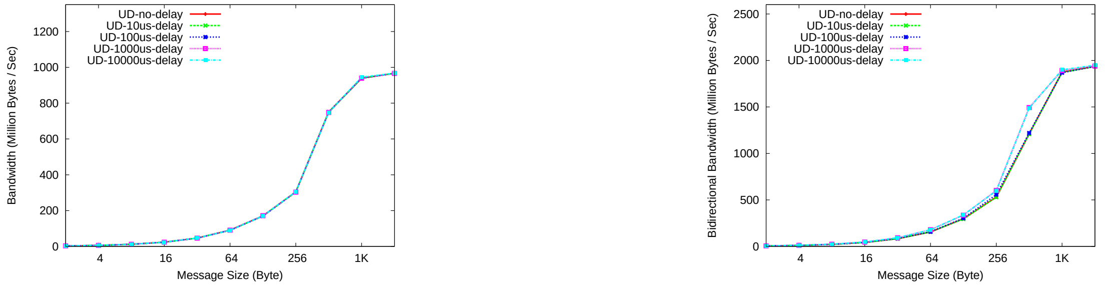

Figure 4. Verbs level throughput using UD: (a) Bandwidth (b) Bidirectional Bandwidth

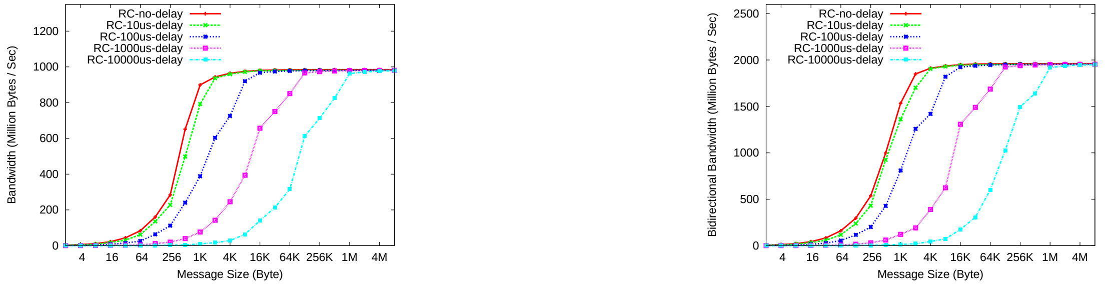

Figure 5. Verbs-level throughput using RC: (a) Bandwidth (b) Bidirectional Bandwidth

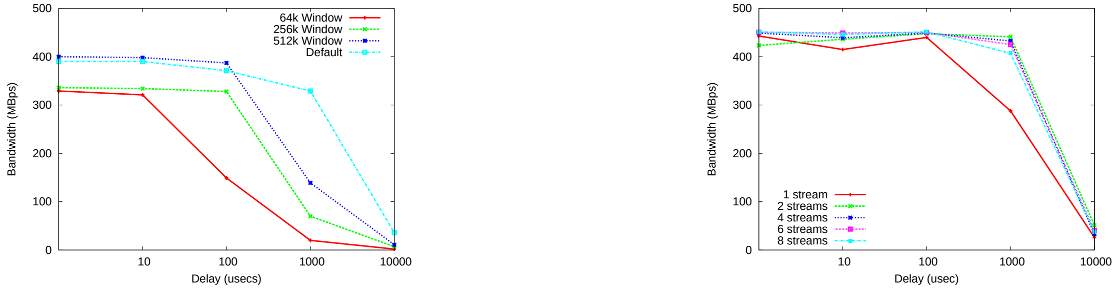

Figure 6. IPoIB-UD throughput: (a) single stream (b) parallel streams

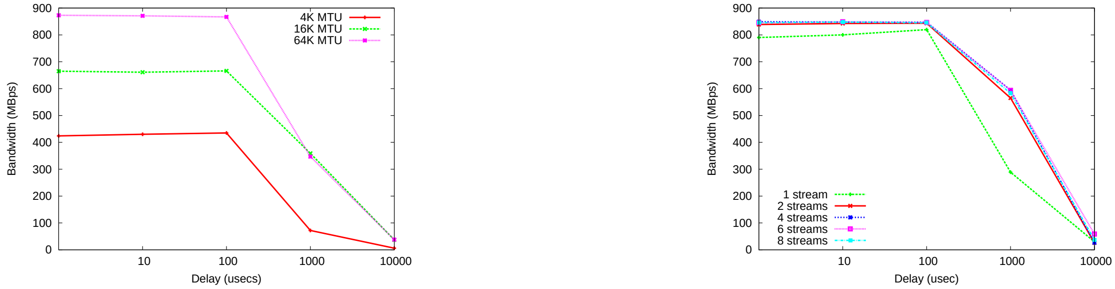

Figure 7. IPoIB-RC throughput: (a) single stream (b) parallel streams

the longer WAN delay (i.e., larger than 100 us) in this case. This drop corresponds to the drop of verbs level bandwidth for 64K message sizes (at 1000us delay) as seen in Figure 5(a) as well.

As in the earlier section, we measure the parallel stream bandwidth of IPoIB-RC. The results are shown in 7 (b). We observe the similar trend that with two or more connections, the bandwidth performance can be better sustained across a wider range of cluster separations. Hence, applications with parallel TCP streams have high potential to maximize the utility of the WAN links.

3.4. MPI-level Performance

In this section, we perform MPI-level evaluations in the various cluster-of-clusters scenarios. In particular, we measure bandwidth performance with increasing inter-cluster delays and present basic optimizations to maximize the obtainable performance. We utilize OSU microbenchmarks (OMB) [15] for all the MPI-level evaluations.

MPI-level Bandwidth: We evaluate MPI communication performance using MVAPICH2 [18] with one communicating process on each of the two clusters. We observe trends similar to the basic verbs-level evaluations with a peak bandwidth of about 969 MillionBytes/sec, as shown in Figure 8(a).

However, the MPI communication protocol utilizes a rendezvous protocol for medium and large message sizes (by default above 8KB for MVAPICH2). This involves an additional message exchange before the actual data-transfer to save the communication buffer copy costs (Zero copy implementations). Due to this we observe that the performance of certain medium size messages is impacted adversely.

Bidirectional bandwidth tests also show similar trends and results are shown in Figure 8(b) with a peak of about 1913 MillionBytes/sec.

Performance Impact of MPI Protocol Tuning: In order to improve the MPI bandwidth performance of medium sized messages, we adjust the MPI rendezvous threshold according to the WAN delay. Figure 9 (b) shows the bandwidth performance of the MPI-level tests running with an emulated network delay of 1ms. The graphs show a a significant performance improvement for certain message sizes with a protocol threshold tuned to 64KB. Bandwidth for a 8KB message size improves by about 44% over the original implementation. Similarly an even more prominent performance improvement of about 83% is seen in the bidirectional bandwidth. Since WAN links are often dynamic in nature, mechanisms like adaptive tuning of MPI protocol, etc. are likely to yield the best performance in normal cases. Also, higher level communication protocols involving additional control messages need to be re-evaluated and adjusted based on the dynamics of the underlying WAN link.

MPI Performance with Multiple Streams: In order to maximize the utilization of the WAN links, in this section we evaluate the performance of MPI with multiple communicating streams. In this test, processes from *Cluster A* communicate with a corresponding process in *Cluster B* forming multiple pairs of communicating processes. The aggregate messaging rate across all these processes is reported.

As shown in Figure 10, the messaging rate grows proportionally to the number of communicating streams for small messages. While we see that a single communicating stream by itself does not perform well under high network delays, multiples of these streams can be combined to achieve a significantly higher aggregate messaging rate across the WAN link. We further observe that for higher delay networks, the additional parallel streams can improve the messaging rate of even medium sized messages. i.e. for higher delay networks, more parallel streams are better for overall network bandwidth utilization.

MPI Broadcast Performance: Collective communication is an important aspect in MPI design. In this section, we optimize MPI broadcast as an example to illustrate the potential benefits of WAN aware communication operations. We used two sets of 64 processes (with 32 nodes with 2 processes on each) on each cluster connected over WAN. We present a simple optimized broadcast (as in [13]) which performs the bcast operation hierarchically over the two connected clusters, minimizing the traffic on the WAN link.

In the *OSU_Bcast* benchmark, the root process sends a broadcast message, using an MPI_Bcast operation, to all the processes in the communication world and waits for an ack from the process with the greatest ack time. Once the root gets the ack from the process with the greatest ack time, it moves on to the next broadcast operation. All the other processes in the communication world will continually wait on the MPI_Bcast collective call. The process with the greatest ack time, which we select beforehand, will send an ACK message back to the root using MPI_Send once it comes out of the collective call so that the root can proceed to the next broadcast operation.

Figures 11 (a), (b) and (c) illustrate the comparison for the latency of the original benchmark and the modified benchmark with 10 us, 100 us and 1000 us WAN delays respectively. We see that the modified algorithm achieves much lower latency for the medium and large messages. For the block distribution mode of MPI processes, the improvement is up to 20%, 18% and 90% for the message of 128K in the above scenarios. For the small messages, as the WAN link is able to handle all the traffic, the

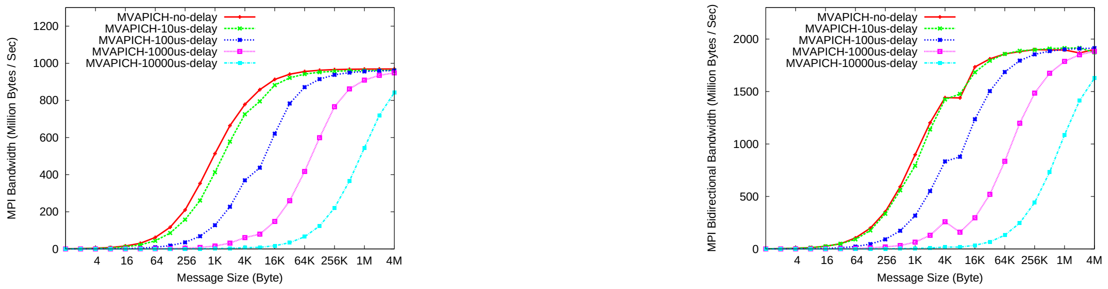

Figure 8. MPI-level throughput using MVAPICH2: (a) Bandwidth (b) Bidirectional Bandwidth

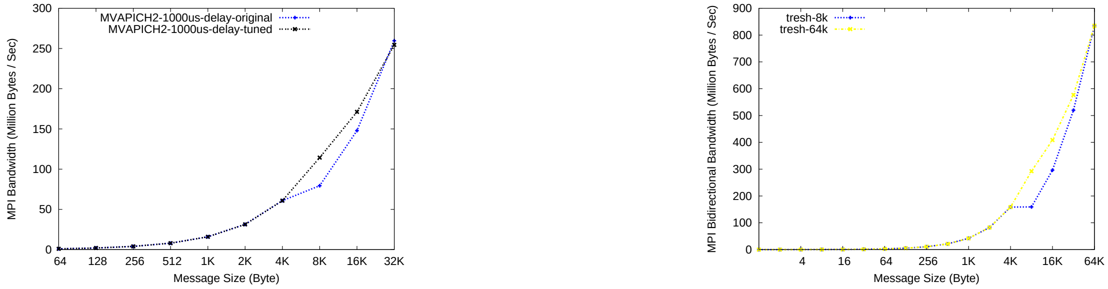

Figure 9. MPI-level throughput using MVAPICH2 with varying protocol thresholds for 1ms delay: (a) Bandwidth (b) Bidirectional Bandwidth

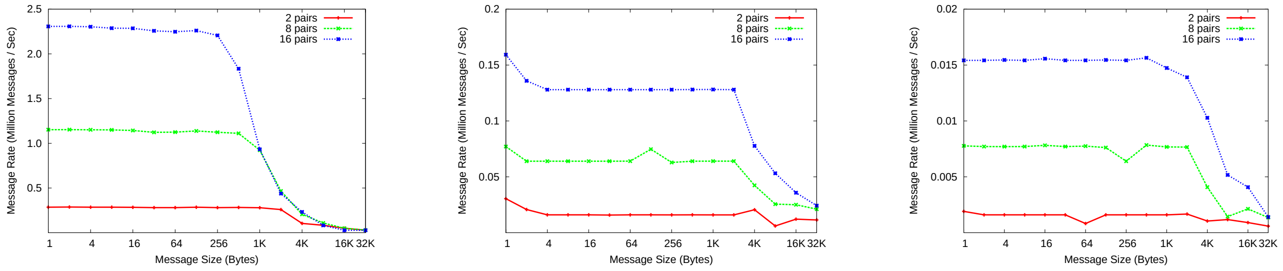

Figure 10. Multi-pair message rate for varying delays: (a) 100us delay (b) 1ms delay (c) 10ms delay

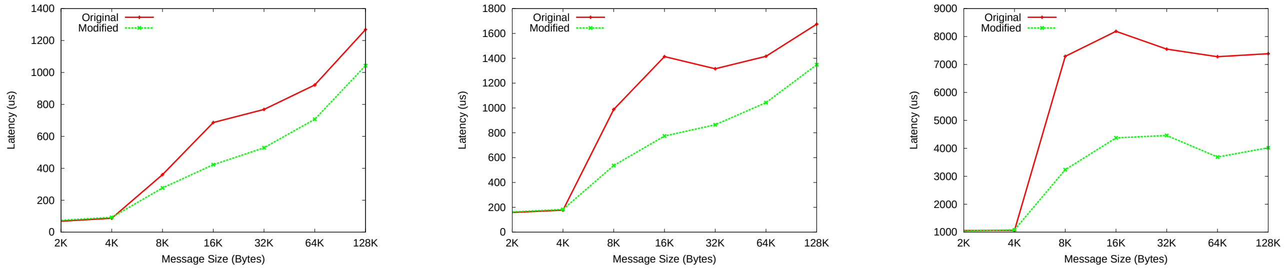

Figure 11. MPI broadcast latency over IB WAN: (a) 10us delay (b) 100us delay (c) 1000us delay

congestion is very minor. We observe that the performance of all the cases is comparable. These results indicate that proper optimizations to the existing collective applications are necessary for scalable performance across IB WAN networks.

3.4.1 MPI Application-level Evaluation

In this section we evaluate the performance of NAS MPI applications running in various cluster-of-clusters scenarios.

Figure 12 shows the performance of NAS [9] class B benchmarks with increasing inter-cluster delay. In this experiment we have 32 processes running on each of the two clusters. We observe that the IS and FT benchmarks show significant tolerance towards added network delay. i.e. our results show that IS and FT application benchmarks can deliver the same performance in the scenario with a separation distance of up to *200 km* as that in the scenario with *0 km* separation. On the other hand, we see that other benchmarks such as CG show a marked degradation in performance for higher network delays.

It is to be noted that the performance of applications running across clusters largely depends on the communication pattern of the application itself and the results we observe for IS, FT and CG are a reflection of their individual communication characteristics. We actually profiled the message size distribution in these applications. It shows that IS and FT involve a high percentage (i.e., 41% and 83% respectively) of large messages while CG has a high percentage of small and medium messages (i.e., all the messages are smaller than 1M). This is essentially one of the main reasons for their performance as can be expected according to the bandwidth results shown in Figure 8.

While the performance observed in these cases is naturally dependant on the specific application being run, it is also important to note that HPC applications (such as IS and FT) seem to tolerate small network delays well in cluster-of-clusters scenarios. Further, with the advent of low-overhead IB WAN networks, cluster-of-clusters has emerged as a feasible architecture for HPC systems.

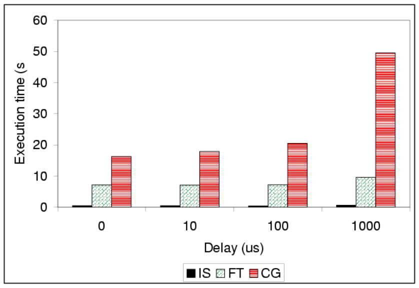

Figure 12. Performance of NAS Benchmarks

3.5. NFS Performance

In this section we use the popular file system benchmark - IOzone [1] to evaluate the throughput of NFS over WAN. We use the single server multi-threaded client model and compare NFS read bandwidth of the NFS/RDMA (as described in Section 2.3) implementation with the regular NFS implementation over IPoIB (for simplicity, we use NFS/IPoIB here on) for varying router delays. NFS Write shows similar performance and due to space constraints, those results are omitted. A 512 MB file with record size of 256 KB is used for all experiments.

NFS/RDMA performance: Using single connection with multiple client threads, we measure the NFS/RDMA read throughput over LAN and over WAN with varying delays and illustrate the results in Figure 13(a). Comparing with the LAN throughput, the introduction of WAN routers degrades the performance by around 36%. This is due to the fact that the WAN speed of IB is at SDR (10Gbps) rates as compared to the DDR (20Gbps) speeds seen in LAN. For WAN scenario, we also see that peak bandwidth with 0 us and 10 us delay is around 700 MB/s while at 100 us delay it drops to 500 MB/s and at 1000 us delay it has a sharp drop to 100 MB/s. Considering that in NFS/RDMA design, the data is fragmented into 4K packets for transferring, these trends are consistent with Figure 5. i.e. the bandwidth of a 4K message drops with larger delays and drops significantly with 1000 us delay.

NFS/RDMA vs NFS/IPoIB: In this experiment, we compare the performance of NFS/RDMA and NFS/IPoIB over WAN. Figures 13 (b) and (c) show the comparison with 10 us delay network and 1000 us delay network respectively. In Figure 13(b), we observe that NFS/RDMA outperforms RC-based NFS/IPoIB by 40% and UD-based NFS/IPoIB by 250%. This is because of the absence of additional copy overheads and lower CPU utilization in the NFS/RDMA design. As seen in Section 3.3, IPoIB-RC shows better bandwidth than IPoIB-UD for NFS operations as well. Further, we observe that for larger delays (Figure 13(c)) NFS over IPoIB-RC does the best. This is again due to the fact that among IPoIB-RC, IPoIB-UD and RDMA(RC) of 4KB, IPoIB-RC gives the best bandwidth for larger delays as seen in previous sections.

4. Related Work

Intel has recently introduced optical cables [5] with CX4 interfaces to extend the reach of InfiniBand. These cables provide very low latency and high bandwidth (DDR) links, enabling InfiniBand connectivity across clusters within 100 meters range. However, clusters separated by more that that length cannot be connected using these. Recently, Bay Microsystems [16] has also announced a IB long-haul

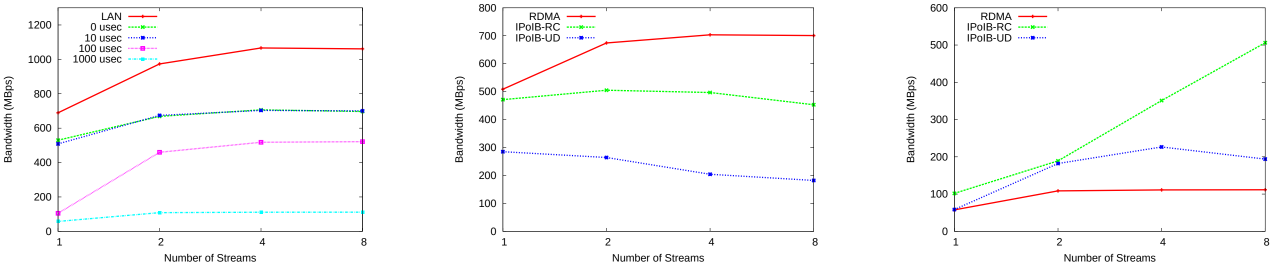

Figure 13. NFS Read throughput of (a) NFS over RDMA, (b) NFS over RDMA and IPoIB with 10 us delay and (c) NFS over RDMA and IPoIB with 1000 us delay

technology that is similar to the Obsidian product.

S. Eikenberry et al. [11] conducted a series of experiments using several grid computing applications (e.g., Linpack, WRF, HOMME, GAMESS etc.) within a campus grid area, and derived the conclusion that Campus Grids created through the linked InfiniBand networks could increase total throughput while bringing relatively simple administrative overhead. Researchers in [6] have evaluated the performance of basic communication primitives and those of file systems like Lustre using InfiniBand WAN technologies over DOEs Ultra-Science Net experimental network. Authors in [19] also characterized the Obsidian Longbows using several techniques and protocols, i.e.1, TTCP over SDP/IB, MPI over IB/VAPI and iSCSI over SDP/IB, demonstrating that the Longbows are capable of high wire speed efficiency. Our work is different from [6] and [19] in that we perform our evaluations in a highly fine-grained way (i.e., we measure the performance with increasing WAN delays). Furthermore, in each of the experiments we propose possible optimizations (e.g., protocol threshold tuning, using parallel streams) and evaluate the improved performance as well. Therefore, our paper is complementary to the existing research and provides more implications to the design and deployment of IB WAN systems in a wide range of cluster-of-clusters scenarios.

5. Conclusions

Trends in High Performance Computing (HPC) systems' requirements coupled with the rapid strides in technology growth at affordable costs have lead to the popularity and wide scale deployments of high performance clusters with modern interconnects like InfiniBand (IB). Further, several large organizations are finding themselves with multiple clusters due to logistical constraints like power/cooling limitations, space constraints, etc. While IB enables HPC applications within individual cluster, performance of HPC applications and middleware across clusters has often been constrained.

To address this issue, IB has recently extended its physical reach with long-haul WAN-capable IB routers, thereby, providing for basic IB-level connectivity across different clusters. However, IB applications, middleware and protocols were all developed under assumptions based on the normal intra-cluster IB characteristics and long-haul IB characteristics can vary drastically from these, leading to severe performance penalties.

In this paper, we have evaluated the following HPC middleware: (i) IPoIB, (ii) MPI and (iii) NFS over RDMA, using Obsidian Longbow IB WAN routers in different cluster-of-clusters scenarios. Our results have shown that applications usually absorb smaller network delays fairly well. However, many protocols get severely impacted in high delay scenarios. Further, we have shown that communication protocols can be optimized for high delay scenarios to improve the performance. Our experimental results show that optimizing communication protocols (i.e. WAN-aware protocols), transferring data using large messages, using parallel data streams (upto 50% improvement for high delay networks) and hierarchical collectives (upto 90% improvement for high delay networks) improved the communication performance in high delay scenarios.

Overall, our results have demonstrated the feasibility of utilizing long-haul IB WAN technology as an inter-cluster interconnect, enabling the use of cluster-of-clusters architectures for HPC systems. As future work we plan to study collective communication operations in cluster-of-clusters scenarios in detail. We further plan to study the benefits of IB range extension capabilities in other contexts including parallel file-systems and data-centers and propose possible optimizations.

References

[1] Iozone filesystem benchmark. http://www.iozone.org.

[2] IPoIB: InfiniBand Linux SourceForge Project. `http://infiniband.sourceforge.net/NW/IPoIB/overview.htm`.

[3] Nfs version 4 protocol specification. http://tools.ietf.org/html/rfc3530.

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

[5] Intel Connects Cables. http://www.intel.com/design/network/products/optical/cables/index.htm.

[6] S. Carter, M. Minich, and N. Rao. Experimental Evaluation of InfiniBand Transports over LAN and WAN Networks. In *High Performance Computing*, 2007.

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

[8] Obsidian Research Corp. http://www.obsidianresearch.com/.

[9] D. H. Bailey and E. Barszcz and L. Dagum and H.D. Simon. NAS Parallel Benchmark Results. Technical Report 94-006, RNR, 1994.

[10] Open Fabrics Enterprise Distribution. http://www.openfabrics.org/.

[11] Steffen Eikenberry, Karl Lindekugel, and Dan Stanzione. Long Haul InfiniBand Technology: Implications for Cluster Computing . http://www.obsidianresearch.com, 2006.

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

[13] P. Husbands and J. C. Hoe. MPI-StarT: Delivering Network Performance to Numerical Applications . In *Proceedings of IEEE/ACM Supercomputing*, 1998.

[14] Jiuxing Liu, Jiesheng Wu, Sushmitha P. Kini, Pete Wyckoff, and Dhabaleswar K. Panda. High Performance RDMA-Based MPI Implementation over InfiniBand. In *17th Annual ACM International Conference on Supercomputing*, June 2003.

[15] OSU Microbenchmarks. http://mvapich.cse.ohio-state.edu/benchmarks/.

[16] Bay microsystems. http://www.baymicrosystems.com/.

[17] Ranjit Noronha, Lei Chai, Thomas Talpey, and Dhableshwar K.Panda. Designing NFS with RDMA for Security,Performance and Scalability. In *Int'l Conference on Parallel Processing (ICPP)*, 2007.

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

[19] Craig Prescott and Charles A. Taylor. Comparative Performance Analysis of Obsidian Longbow InfiniBand Range-Extention Technology. http://www.obsidianresearch.com, 2007.