Comparative Performance Analysis of Obsidian Longbow InfiniBand Range-Extension Technology

Craig Prescott and Charles A. Taylor
High-Performance Computing Center
University of Florida
Gainesville, Florida 32611

***Abstract*— In January 2007 the UF HPC Center placed into production a Thirty-three-terabyte storage subsystem based upon Rackable Systems' RapidScale clustered file system. This subsystem utilizes Sockets Direct Protocol over an InfiniBand interconnect (SDP/IB) for iSCSI transport and has sustained in excess of 3.2 GB/s of aggregate throughput for random write access patterns. For remote clusters within our campus grid to access this storage using native protocols it was necessary to extend the reach of our InfiniBand (IB) fabric. In this paper we present the results of I/O tests run from remote cluster nodes to the storage subsystem using Longbow IB Range Extension technology from Obsidian Research. For comparison, we also present the results of network-only (memory-to-memory) throughput tests.**

***Index Terms*—InfiniBand, Range Extension, SDP, Socket Direct Protocol, iSCSI, Storage**

I. Introduction

Over the last ten years advances in microprocessor architectures, system interconnects, and parallel programming APIs have given rise to large, commodity-based, high-performance computing systems popularly referred to as "clusters". Such distributed-memory, parallel computing systems can attain impressive computational performance on the order of tens to hundreds of teraflops. Due to the economies of scale inherent in the production of the components from which such clusters are built, the cost per unit of computational capacity is substantially lower than that of monolithic, single-system-image machines. Hence, the emergence of clusters in the HPC arena where academic institutions and government agencies are perpetually short of funds is not surprising.

Charles A. Taylor. Jr. is the Associate Director of the High-Performance Computing Center at the University of Florida, Gainesville, Florida 32611 USA (Phone: 352-392-4036, e-mail: taylor@hpc.ufl.edu).

Craig Prescott is a Senior Scientist with the High-Performance Computing Center at the University of Florida, Gainesville, Florida 32611 USA (Phone: 352-392-2773, e-mail: prescott@hpc.ufl.edu).

Just as microprocessors and specialized communication ASICs (Application Specific Integrated Circuits) have increased in power and performance in a manner roughly consistent with Moore's Law[1], so too has the areal density of magnetic storage media. From the introduction of the magnetic disk in 1956[2] to now, capacities have increased from 1.768×10^3 bits/sq. in. to 1.28×10^{11} bits/sq. in. - current capacities of a single drive are at 750 GB (giga=10^9) today. Hence capacities have increased at a 43.61% annual growth rate for 50 years and are now seventy-two million times greater than in 1956.

However, overall data transfer rates have not kept pace. By comparison, today's fastest fibre channel disk drives are roughly twenty-three thousand times faster than the IBM 350 Disk File introduced in 1956. This represents an annual increase of only 22.23%. Hence, the cost per unit of storage capacity has dropped dramatically but the ability to access that storage without crippling computational performance across a large cluster remains. Building a cost-effective I/O subsystem that can keep pace with today's clusters remains a daunting challenge.

RAID[3] technology has provided a partial solution in that hardware-based striping enables the simultaneous use of multiple storage devices. This has worked well for increasing data transfer rates on single-instance servers where the RAID hardware presents a single logical unit (LUN) to the host system while parallelizing block-level transfers to and from the LUN. This means that a single host, namely the directly attached server, can read and write data at rates that are many multiples (stripe-depth) of the single disk transfer rates. But distributing this parallel I/O capability to one or more nodes in a cluster is hampered by limited network bandwidth, cumbersome communication protocols, and inefficient allocation of data among multiple servers.

For many years the *de facto* standard for sharing data among two or more computers on a network has been the Network File System (NFS) introduced by Sun Microsystems, Inc. in 1985. Despite its success and usefulness, NFS suffers from several deficiencies as a shared file system.

1. Clients are not guaranteed a consistent view of the same file data (i.e. there is no local cache coherency among clients).
2. A single file system cannot be distributed consistently by more than one server (i.e. it is not parallel and does not scale).
3. The high overhead and low throughput associated with the protocol itself as well as the underlying protocols (TCP/IP,UDP) diminish performance.
4. File and byte range locking that has been historically buggy and unreliable.

These deficiencies have combined with the ever-growing size and power of clusters to create a need for a true parallel, cluster file system that can provide a) a global, unified name space; b) local cache coherency; c) global lock management services; d) scale with the addition of storage and server resources.

A number of such file systems are now commercially available. Among these are Lustre (Cluster File Systems, Inc.), PanFS (Panasas), FusionFS (IBRIX), GPFS (IBM) and RapidScale (Rackable Systems, Inc.). We make no attempt here to provide any kind of comprehensive review or comparison of these competing products. Rather, we simply note the performance achieved with our choice of file system (RapidScale) and use it to demonstrate the use of Obsidian Research Corp.'s Longbow InfiniBand Range-Extension technology.

II. Test Configurations

The test environment consisted equipment in two buildings, the High-Performance Computing (HPC) Center in the Physics building and the High-Performance Computing and Simulation Research (HCS) Lab in Larson Hall. The RapidScale file system and associated InfiniBand fabric, storage arrays, iSCSI targets, and iSCSI initiators were located at the HPC Center. The HCS Lab had an InfiniBand fabric and iSCSI initiators (clients) only. The topology of the experiment can be seen in Figure 1. A more detailed description of each of the components of the experiment follows below.

A. Storage

Storage consisted of twelve OmniStor 4932F Fibre Channel (FC) RAID controller shelves. Each RAID controller had 1024MB of on-board cache, two 2Gb/s FC host interfaces and managed twelve disks (Seagate ST3300007FC). The disks were configured into two RAID-5 logical disks each having six physical disks, a 128 KB stripe depth (768 KB stripe width). Each array was configured with 16MB of write cache and "automatic" allocation of read cache.

B. I/O Servers

A total of eight Rackable Systems C3001 servers were used to host the storage. Each server was a four-way SMP host with two AMD Opteron 275 (2.2GHz, Dual-Core) CPUs, 1 MB of level-two cache (per core), and 4 GB of 400MHz DDR SDRAM on a Tyan K8SE (Thunder) system board. The storage interface consisted of three 2Gb/s FC interfaces (1

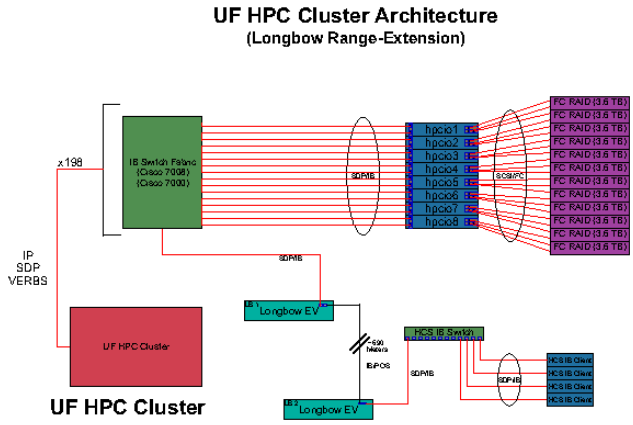

Fig. 1. Architecture diagram for the InfiniBand range-extension experiment.

QLogic QLA2340, 1 QLogic QLE2362) installed in a PCI-X slot (66/133MHz) and PCI-E slot (x4), respectively. Additionally, each server was equipped with a Cisco 4X IB Host Channel Adapter (HCA) (dual-port, LionCub). The HCAs were flashed with version 4.7.600 of the firmware for the 3.2.0.83 Cisco host software release. The operating system software environment was CentOS release 4.0 upgraded to run the Linux 2.6.9-34 ELsmp (x86_64) kernel.

The FC interfaces were cabled to the RAID controllers such that each LUN (logical disk) had a dedicated (200 MB/s) FC link from the host to the controller. Hence, each server hosted three LUNs with an aggregate *potential* I/O throughput of 600 MB/s. For manageability, each LUN was encapsulated as an LVM-2 logical volume prior to file system formatting. The unformatted capacity of each LUN was 1.62 TB. The LVM configuration is shown in Table 1.

TABLE I
Representative I/O Server LVM Configuration

```
hpcio5# pvscan
  PV /dev/sdc   VG R6L0   lvm2 [1.62 TB / 0    free]
  PV /dev/sdb   VG R7L0   lvm2 [1.62 TB / 0    free]
  PV /dev/sda   VG R6L1   lvm2 [1.62 TB / 0    free]
  Total: 3 [4.86 TB] / in use: 3 [4.86 TB] / in no VG: 0 [0]

hpcio5# vgscan
  Found volume group "R6L0" using metadata type lvm2
  Found volume group "R7L0" using metadata type lvm2
  Found volume group "R6L1" using metadata type lvm2

hpcio5# lvscan
  ACTIVE  '/dev/R6L0/IO5-R6L0' [1.62 TB] inherit
  ACTIVE  '/dev/R7L0/IO5-R7L0' [1.62 TB] inherit
  ACTIVE  '/dev/R6L1/IO5-R6L1' [1.62 TB] inherit
```

C. Clients

The HPC I/O clients consisted of varying numbers of Rackable Systems C1000 (1U) servers. Each server was a four-way SMP host with two AMD Opteron 275 (2.2 GHz, dual-core) processors, 1 MB of L2 cache (per core), and 4 GB

of 400MHz DDR SDRAM on a Tyan K8N-DRE system board. Additionally, each server was equipped with a Cisco 4X IB HCA (dual-port, LionCub). The HCAs were flashed with the version 4.7.600 firmware for the 3.2.0.83 Cisco host software release. The operating system software environment was CentOS release 4.0 upgraded to run the Linux 2.6.9-34 ELsmp (x86_64) kernel.

The test environment at the HCS Lab was comprised of four I/O client nodes connected to a Cisco SFS 7000 InfiniBand (IB) switch. Each of the HCS I/O clients had two AMD Opteron CPUs (1.4 GHz and 2.2 GHz, see below) with 1MB of L2 cache and a Voltaire HCA400 InfiniBand HCA installed in a 133 MHz PCI-X slot on a Tyan Thunder K8S (S2880) system board. The IB HCAs were flashed with the Cougar firmware included in the 3.2.0.83 Cisco host software release. The operating system installed on the HCS I/O client nodes was provided by the Rocks 4.1 Linux distribution, configured as stock compute node appliances and upgraded to run the Linux 2.6.9-34 ELsmp kernel.

The HCS I/O client nodes were tested in two configurations. The first, which we subsequently refer to as HCS(240), consisted of a pair of AMD Opteron 240 processors (1.4 GHz, single-core) and 1 GB of ECC DDR333 RAM. In the second configuration, referred to as HCS(248), each HCS client was upgraded to AMD Opteron 248 processors (2.2 GHz, single-core) and 2 GB of RAM.

D. Interconnect (InfiniBand)

Each of the I/O servers (iSCSI targets) and clients (iSCSI initiators) was connected to an InfiniBand (IB) fabric consisting of two Cisco SFS 7008 "core" IB switches (72 ports each) and fourteen Cisco SFS 7000 "leaf" switches (24 ports). The leaf switches were connected to the core switches in a Constant Bisectional Bandwidth[4] (CBB) or "Fat Tree" topology. Each leaf switch hosted sixteen clients and had eight ports allocated for inter-switch links (ISL) to the core switches to form a 50% blocking, two-tiered CBB network.

E. Parallel File System (RapidScale[5])

RapidScale™ is a parallel or "cluster" file system product from Rackable Systems, Inc. that provides 1) a global name space, 2) local cache coherency, and 3) a global locking service. It is a client-server architecture in which the client-side consists of a loadable kernel module implementing a version of the iSCSI protocol modified to include RapidScale's cache coherency logic. The server side consists of a user-space daemon that plays the role of an iSCSI target managing data and meta-data requests while enforcing cache coherency and performing lock management.

In the RapidScale test configuration, each I/O server or "target" hosted three XFS file system containers atop each of the three LUNs (described previously) with a RapidScale target software daemon for each container. Thus there were a total of twenty-four RapidScale containers each with their own server process distributed among eight servers. This configuration sustained over 3.2 gigabytes per second of random-write throughput from sixteen IOzone[6] threads running on four clients through the local InfiniBand fabric.

F. InfiniBand Range-Extension (Obsidian Research)

The RapidScale file system described above was implemented using SDP/IB as a transport layer for the iSCSI protocol. By choosing RapidScale over SDP/IB as the UF HPC Center's cluster file system, we were able to achieve much greater aggregate and per-client I/O rates to the file system than would have been possible using TCP/UDP over gigabit Ethernet. However, this choice also limited the availability of the file system to clients directly attached to the local IB fabric, as the UF HPC InfiniBand infrastructure is built upon 4X SDR copper links - RF losses limit the maximum range of such links to less than 20 meters. Since our goal was to distribute the RapidScale file system to remote clusters on the UF campus, we needed a way to extend the range of our IB fabric.

Obsidian Research Corp. has developed the technology to extend the range of an InfiniBand link over conventional campus-area and wide-area networks. This technology has been incorporated into a series of products under the name "Longbow". Each Longbow has a 4X SDR copper InfiniBand port for connecting to an IB-enabled host or fabric and an optical port to connect to existing networks or dedicated fiber. The Longbows encapsulate IB traffic in a variety of protocols for transport over the WAN via the Longbow's optical port. Thus, the topology for extending an InfiniBand fabric requires a pair of Longbow units joined by a fiber path – one Longbow connected directly to the IB fabric at the "near" end, and the other at a fabric at the "far" end. Each Longbow looks like a 2 port IB switch to the subnet manager, and the fabrics become merged into a single subnet. Since all InfiniBand semantics are preserved, the extension of the IB fabric is totally transparent at all levels. However, this is not enough.

Though the fiber connection avoids the distance limitations imposed by copper cables at the physical layer, the buffers used to manage IB flow control at the link layer are small (optimized for short signal flight time). This limits the effective range to a few hundred meters. InfiniBand flow control is based upon "credit"; at initialization, each endpoint on an IB fabric declares its capacity to receive data (its "buffer credit") and continually notifies the fabric of updates to this capacity as buffers are freed. As distances become greater than a few hundred meters, receive credits cannot be restored fast enough to keep the pipe full. Thus, throughput falls as latency increases.

The Longbows provide a solution to the range limitations imposed at the link layer via buffer credit extension. Each Longbow provides enough memory (buffer credit) and the processing power to manage the buffer credit extension to function as a sort of "buffer credit middle-man", and is capable of keeping the pipe full by restoring the receive

credits faster. Thus, full InfiniBand performance is thereby restored over long distances, without application-level flow control concerns. Different Longbow models provide capability to extend IB fabrics to varying ranges. The Longbow XR is designed for ranges over 100,000km, and has been publicly tested over 14,000 km OC-192c networks, achieving near wire-speed from single workflows [7]. Longbow Campus is optimized for direct optical connections up to 10km in length, and is more suitable for dedicated fiber links between buildings.

The distances of interest at the University of Florida are not quite so global; 10km is more than enough to connect clusters of significant computing capacity distributed across campus. To that end, Obsidian Research sent two evaluation prototypes to the UF HPC Center in August of 2006.

G. *Procedures*

Latency, network throughput, and storage throughput over the Longbow links was measured in several test configurations using TTCP[8] over SDP (TTCP/SDP), IMB (Intel MPI Benchmarks)[9], and IOzone. Tests were run over a) 3 meters via a single-mode fibre patch cable, b) 2000 meters via a fiber spool simulator [10], and c) 690 meters via dedicated fiber between the HPC Center machine room and the HCS lab. In some cases tests were run over our conventional IB fabric with no Longbows in the communication path. The configuration of the Longbow units was minimal for links over dedicated fiber and consisted of setting the WAN ports to use Packet-Over-SONET (POS).]

Latency measurements were made with IMB utilizing MVAPICH[11] 0.94 as distributed by Cisco with the IB host software stack. Mpi_latency, an additional utility also included with the Cisco host stack was also used. Network throughput was measured using TTCP/SDP as well as IMB. Note that the TTCP/SDP measurements are particularly relevant given that the RapidScale storage software also uses SDP. Mpi_bandwidth, similar to mpi_latency, was used as an additional test. Storage throughput was measured with IOzone. Random write tests were performed across a range of block sizes. The random write tests were chosen to minimize spindle effects due to the presence of write back cache on the raid controllers while still accurately representing the I/O patterns of the UF HPC Center users.

Although our primary interest was in storage throughput, the purely network throughput tests were deemed useful to understanding the effects of the Longbow links on SDP/IB throughput without the added complications of disk I/O. Finally we note that functionality provided by a pair of fiber-connected Longbows can be thought of as a single InfiniBand cable, albeit one that could be many kilometers long. Therefore the latency and throughput characteristics of Longbow-connected IB links are most naturally and fairly compared to the same measurements performed over a single InfiniBand cable. Thus, Longbow results will be presented along with results of the same tests run over a single, local IB cable.

III. Results

In this section, we present the results of the tests described previously in section II, *Procedures*. In the plots that follow we use the prefix "M" (as in MBytes) to mean 1 x 10^6 and the prefix "Mi" (as in MiBytes) to mean 2^{20} (1024^2) and similarly for K (1 x 10^3) and Ki (2^{10}). Furthermore, we use the notation HCS(240), HCS(248), and HPC to refer to the three test node configurations described in section IIC. Thus, "HCS(248)-Longbow-HPC" indicates that the measurement was performed between an upgraded HCS host and an HPC host connected via a Longbow link. Where applicable, it may be assumed that the left-most node is the sender and the right-most node is the receiver. For MPI-based tests involving heterogeneous nodes the first argument to mpirun was the HCS host, and the second the HPC host. Finally, unless otherwise specified, the distance between Longbows was approximately 690 meters.

A. *TTCP/SDP*

Figures 2 and 3 show unidirectional TTCP/SDP throughput as a function of buffer size between two HPC nodes for one and four tx/rx threads, respectively. Each figure shows three curves. The first represents a run over the conventional HPC IB fabric in the absence of the Longbow units. For the second curve the Longbow units have been added and connected via a 3 m fiber patch cable. The third curve was obtained after replacing the 3 m patch cable with a 2 km fiber spool hence increasing the separation between the Longbow units from 3 m to 2 km. For all tests presented here the data stream transferred between the sender and receiver totaled 16 GiBytes per stream.

Figure 4 shows unidirectional TTCP/SDP throughput between HCS nodes for a single tx/rx pair with and without the Longbows in the data path. Results for both the HCS(240) and HCS(248) configurations are included.

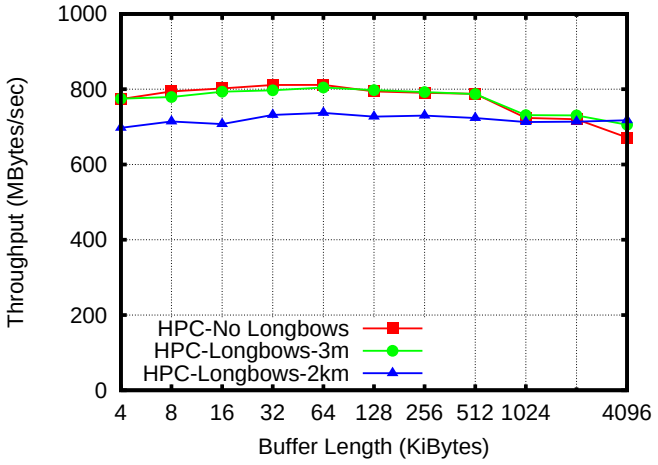

Fig. 2. TTCP/SDP throughput for a single thread between a pair of HPC clients.

Figure 5 shows aggregate TTCP/SDP throughput for eight tx/rx pairs. In this case, each of four HCS nodes ran two tx threads and each of four HPC nodes ran two rx threads. Again, results from both HCS host configurations are shown.

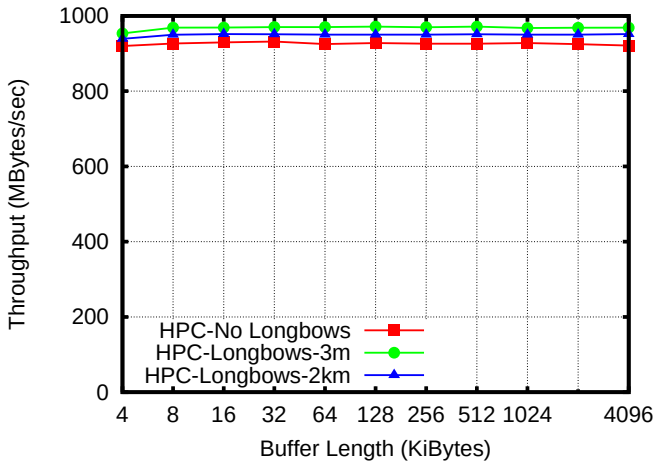

Fig. 3. TTCP/SDP throughput for four threads between a pair of HPC clients.

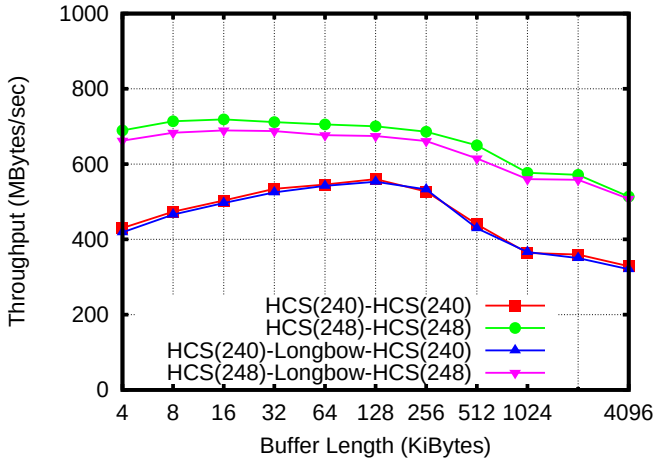

Fig. 4. TTCP/SDP throughput for a single Rx/Tx pair running between two hosts, with and without Longbow links.

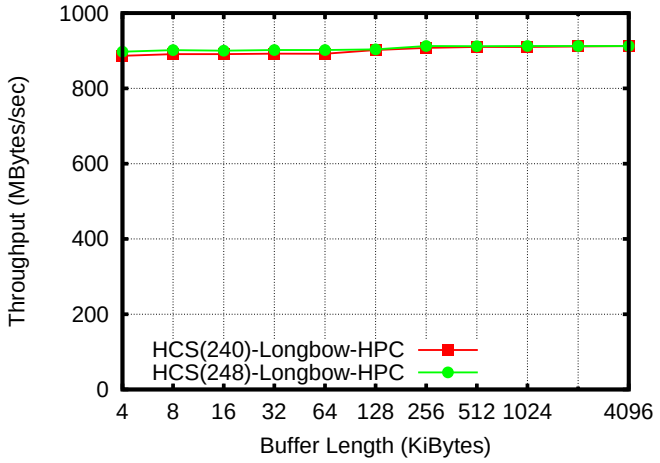

Fig. 5. TTCP/SDP throughput for eight tx/rx pairs running across the Longbows between HCS and HPC clients (two threads per client).

B. *Intel MPI Benchmarks (IMB)*

Figures 6 - 13 show the latency and throughput results as a function of message size for the PingPong, PingPing, SendRecv, and Exchange tests from the Intel MPI Benchmarks (IMB). Results are presented for HPC nodes and both HCS node configurations with and without the Longbows in the data path.

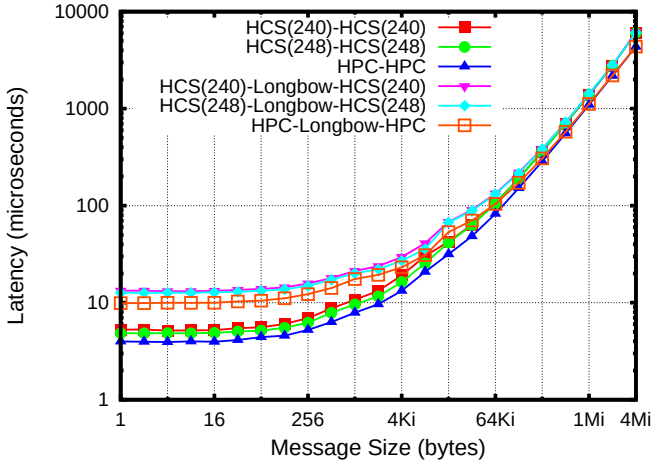

Fig. 6. IMB PingPong Latency

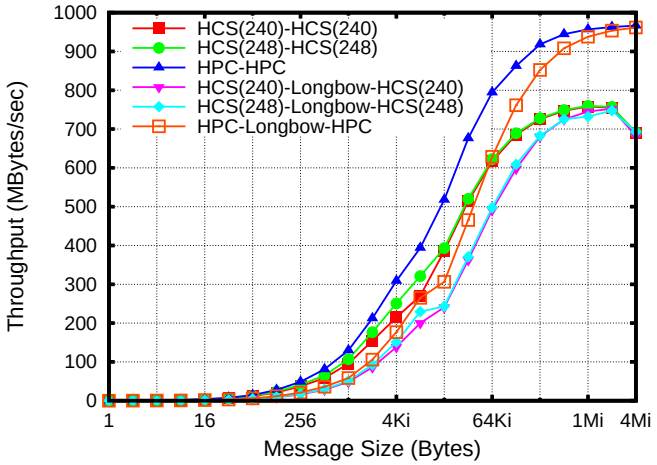

Fig. 7. IMB PingPong Throughput

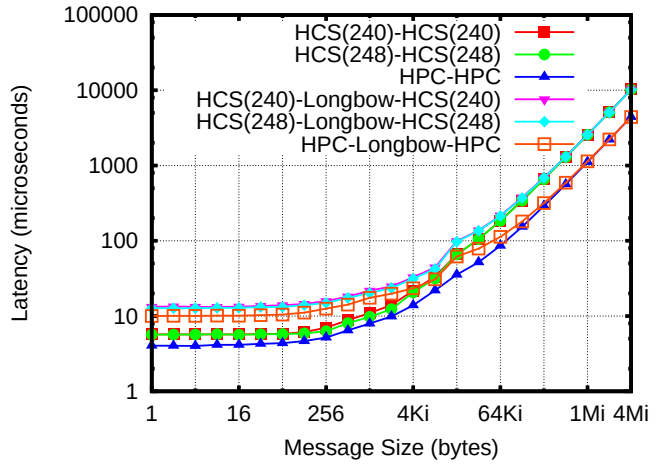

Fig. 8. IMB PingPing Latency

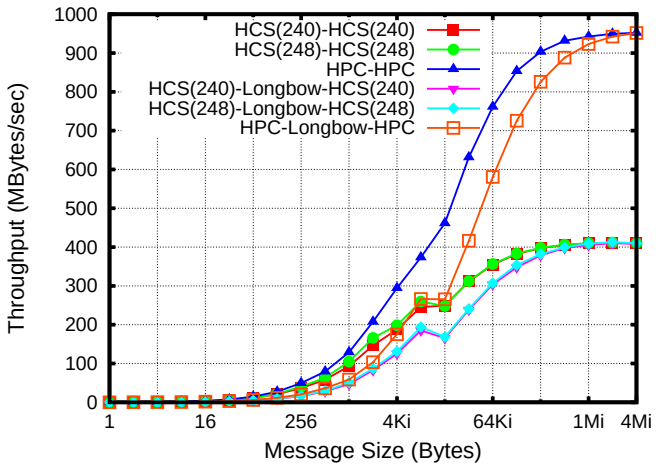

Fig. 9. IMB PingPing Throughput

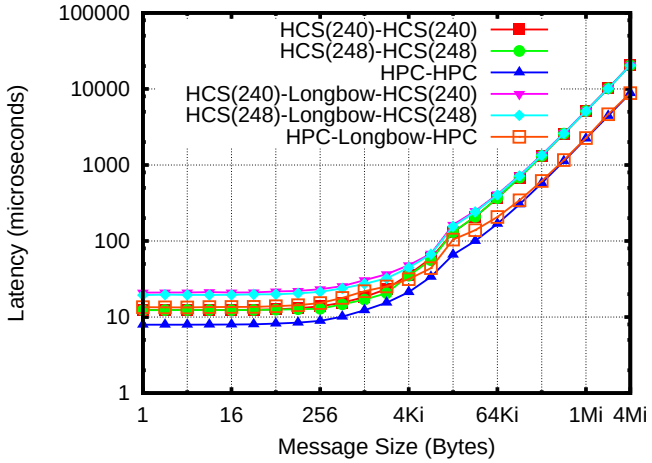

Fig. 10. IMB Exchange Latency

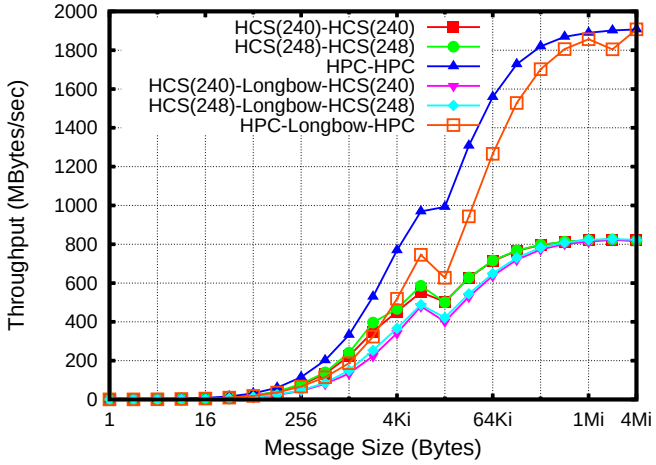

Fig. 11. IMB Exchange Throughput

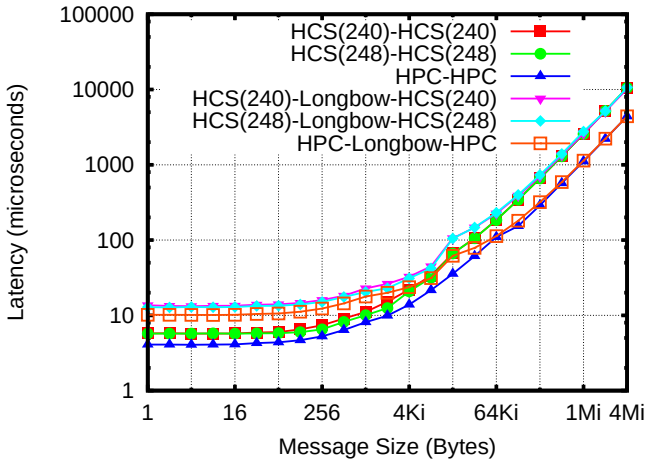

Fig. 12. IMB SendRecv Latency

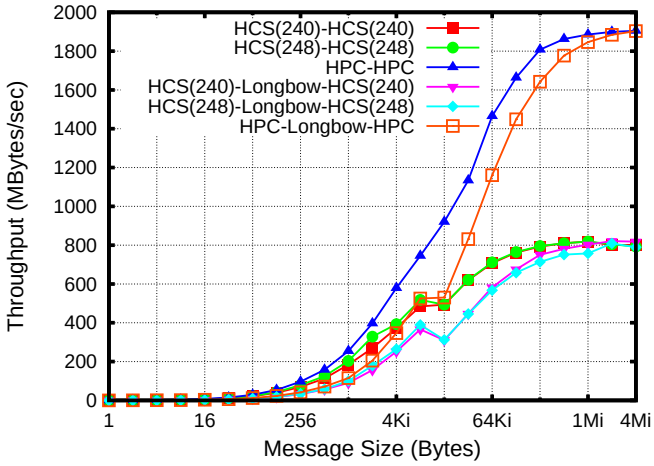

Fig. 13. IMB SendRecv Throughput

C. *MPI Latency*

Figure 14 shows the results of the mpi_latency program included in the Cisco/Topspin stack as a function of message size for HCS nodes and HPC nodes.

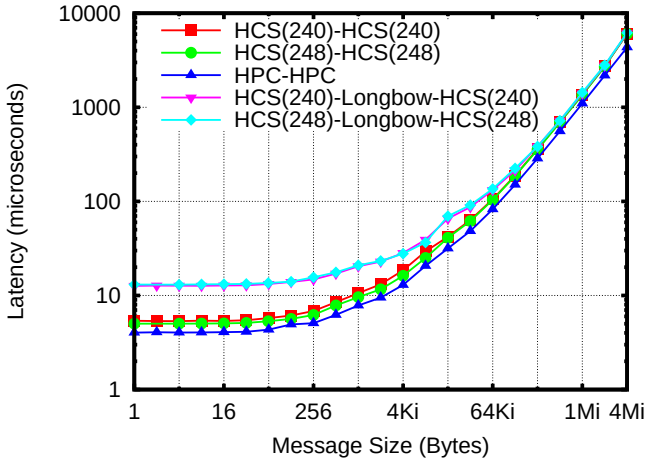

Fig. 14. Mpi_latency latency measurements between HCS and HPC hosts with and without Longbows.

D. *MPI Bandwidth*

Figure 15 shows the results of the mpi_bandwidth program included in the Cisco/Topspin stack as a function of message size for HCS nodes and HPC nodes.

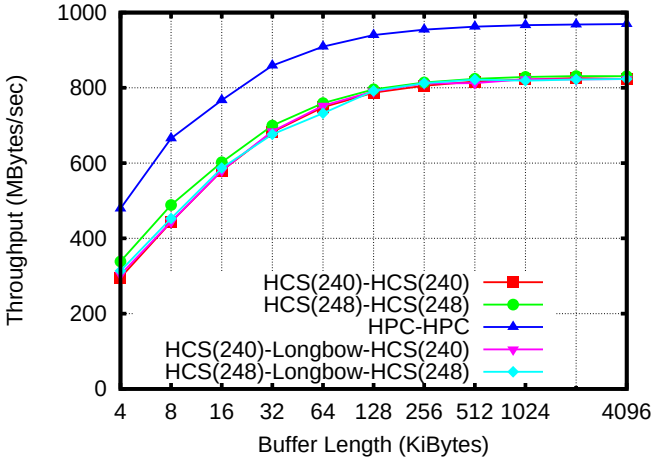

Fig. 15. Mpi_bandwidth throughput measurements between HCS and HPC hosts with and without Longbows.

E. *IOzone*

Figures 16 - 18 show IOzone random write throughput as a function of block size for trials writing to the RapidScale file system built on RAID5 LUNs described earlier.

We were also able to perform IOzone runs using a RapidScale file system built on 12 RAID0 LUNs (4 I/O nodes, 6 RAID controllers). Figures 19 and 20 show IOzone random write throughput as a function of block size for this configuration.

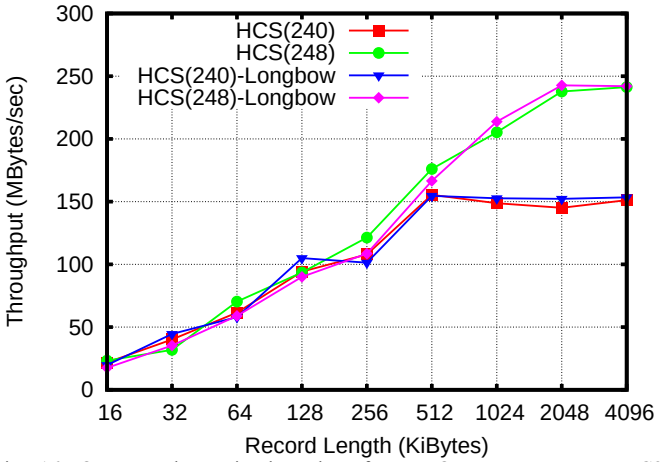

Fig. 16. IOzone random write throughput for one IOzone process on an HCS node with and without Longbows.

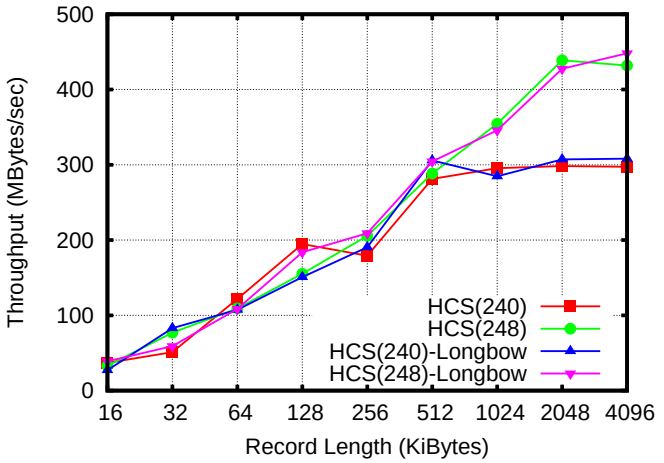

Fig. 17. IOzone random write throughput for two IOzone processes (one per node) on HCS nodes with and without Longbows.

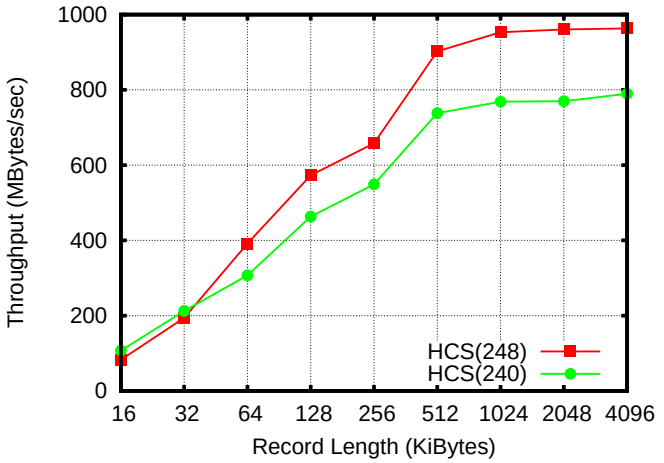

Fig. 18. IOzone random write throughput for eight IOzone processes (one per node) on HCS nodes via the Longbows.

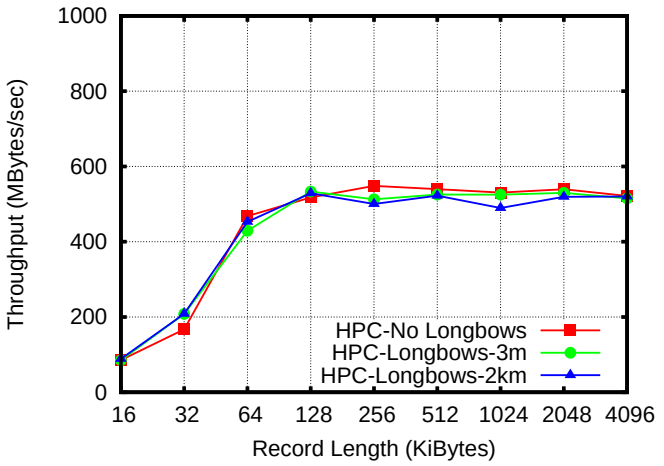

Fig. 19. IOzone random write throughput for four threads from a single HPC client.

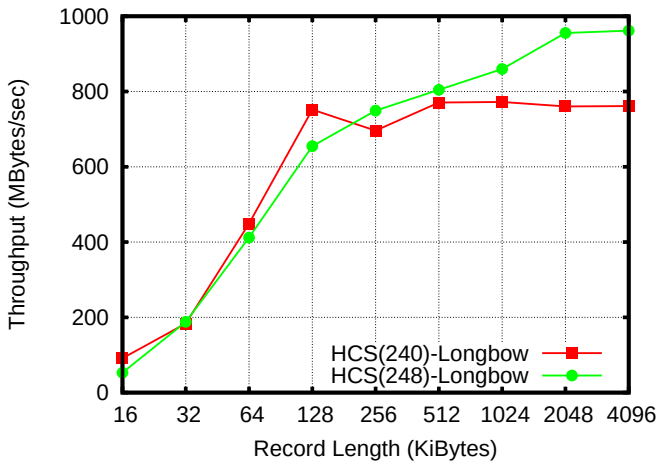

Fig. 20. IOzone random write throughput from four HCS clients each running two threads.

IV. Discussion

A. TTCP/SDP

In Figure 2 it can be seen that the overall throughput through the pair of Longbows connected by a 3 m patch cable is the same or slightly better (see below) than that obtained over a conventional IB fabric/cable. As the distance between the Longbows is increased from 3 m to 2 km, the effects of the additional latency become apparent as overall TTCP throughput is reduced for buffer sizes up to 512 KiBytes. For buffer sizes larger than 512 KiBytes, we observed a marked decrease in throughput in the case where there was no IB range extension, and in the case where the Longbows were connected with the 3 m patch cable. This decrease is a combined result of the Opteron CPU's 1 MB L2 cache and the buffer copy (BCopy) data path implemented within the SDP stack. The BCopy data path requires two buffers to transfer data, one in user space and one in kernel space. Buffers larger than 512 KiBytes cause a flush and reload of the L2 cache in order for the copy to proceed. Others have observed this local cache effect for SDP [12,13,14] as well as with other protocols.

For the case where the Longbows are connected via 2 km fiber spool, the effect of the additional latency is more pronounced – throughput is reduced significantly relative to the other two curves. We initially assumed that the reduced throughput was the result insufficient buffering in the SDP stack. However the default SDP_INET_SEND/RECV_WIN size is 256 KiBytes which is more than enough to accommodate the approximately 60 KiBytes of in-flight data predicted by the bandwidth-delay product (BDP). Further investigation is needed to determine the origin of this reduced throughput in the presence of 2 km fiber spool. We do not observe the cache effect described above in the case of the 2 km spool – the throughput for a single tx/rx pair is never high enough for the cache flushes to become apparent.

Figure 3 shows the aggregate unidirectional TTCP/SDP throughput achieved for four tx/rx pairs (one for each host core) as a function of buffer size for the same three cases as in figure 2. The InfiniBand physical layer uses 8B/10B (IEEE 802.3.z) line encoding. In this scheme there are two bits of signaling overhead for every ten bits of data transmitted leading to an effective data transmission rate of 8 Gbps or a maximum theoretical bandwidth of 1000 MBytes/per second. We can see that the aggregate bandwidth sustained by four tx/rx pairs comes very close to achieving this rate and that there is no significant dependence upon buffer size. It can also be seen that that the maximum achievable throughput is slightly higher in the presence of the Longbows. With the 3m patch cable, the aggregate throughput reaches over 970 Mbytes/sec or approximately 97% of the theoretical bandwidth. The aggregate throughput over the 2 km spool was slightly exceeded 950 MBytes/sec. These rates compare to 920 MBytes/sec without the Longbows. We believe that the additional buffer credits contributed to the IB data-link layer by the Longbows leads to more efficient use of available bandwidth in the presence of multiple data flows.

In figure 4 we see that for a single tx/rx thread TTCP/SDP throughput is dependent upon the speed of the processor and that the HCS(248) node is faster than the HCS(240) node by a factor that is approximately that of the ratio of the processor clock speeds. This implies a high level of processor overhead in the TTCP/SDP stack processing which was apparent as processor utilization approached 100% during these tests. This high level of processor overhead points out a weakness in the SDP stack implementation – namely, the lack of a zero-copy (Zcopy) data path. The work of copying data from user buffers to kernel buffers at 10 Gbps network speeds consumes almost all of an Opteron processor at frequencies below 2.2 GHz. Goldenberg, *et al*. have observed up to an 80% decrease in processor utilization in a preliminary Zcopy SDP stack implementation.

Comparing the HPC single-thread throughput of figure 2 to the HCS(248) single-thread throughput of figure 4 we see that the HPC nodes reach a maximum throughput of a little over 800 MBytes/sec while the HCS(248) nodes peak well below 700 MBytes/sec. This difference is attributable to two factors. The first is the difference in observed [15] InfiniBand performance between PCI-E and PCI-X based host adapters. The second is improvements in the third-generation IB chipset (MT25208) in the HPC host adapters relative to the second-generation chipsets (MT23108) in the HCS host adapters [16].

We also note that for the HCS(248) configuration we see a drop in throughput of 27.6 MBytes/sec across the 690 m distance spanned by the Longbows. In figure 2 we saw an 80.6 MByte/sec drop in throughput across the 2 km distance spanned by the Longbows (through the spool). The ratio of these two distances is 0.34. Likewise, the ratio of the throughput decreases is also 0.34 leading us to conclude that the decrease in TTCP/SDP throughput over the Longbow link is linear with distance, and hence latency. We do not see a similar drop in throughput in the case of the HCS(240) nodes because these nodes are not fast enough to expose this effect.

In figure 5, we see the aggregate unidirectional TTCP/SDP throughput for eight HCS transmit threads (one per processor in each of four HCS nodes) sending to eight HPC receive threads (two per HPC node). Approximately 910 MBytes/sec of aggregate throughput was observed for both HCS node configurations. This result is 60 MBytes/sec less than that achieved between HPC nodes over the 2 km fiber spool. The cause of this difference is left to future investigation.

Finally, we observed in figure 2 that we do not achieve wire speed for a single tx/rx pair with TTCP/SDP, with or without the Longbows. The peak throughput achieved on the local InfiniBand fabric was just over 800 MBytes/sec. However, we note that with the OFED version 1.2 InfiniBand host software stack from the OpenFabrics Alliance[17], a single tx/rx pair between HPC clients achieves unidirectional throughputs over 960 MBytes/sec.

B. MPI Benchmarks

A subset of the Intel MPI Benchmarks (IMB) was chosen to characterize the throughput and latency characteristics of the InfiniBand interconnect with and without Longbow range extension. This subset included PingPing, PingPong, Exchange, and SendRecv. The mpi_latency and mpi_bandwidth tools included in the Cisco/Topspin host InfiniBand software stack were also used. Since the tests were run on nodes of two different architectures at two different CPU speeds, several noteworthy features are observed. First, it can be seen that the overall throughput with MPI protocols of the HCS nodes is capped at roughly 830 MBytes/sec regardless of the CPU speed. This rate is consistent with commonly observed maximum PCI-X throughput rates in PC architectures[18]. Hence we conclude that the maximum InfiniBand throughput for the HCS nodes is limited by the PCI-X bus.

We can also observe a knee or bend in many of the plots in the transition from 8K to 16K message sizes. This boundary marks the default point at which MVAPICH transitions from an eager to rendezvous protocol for message passing. This knee is particularly pronounced in the presence of the Longbows and is a direct consequence of the increased latency across the WAN link. An increase in latency adds to the initial setup overhead when using a rendezvous protocol. For smaller messages, the increased transfer efficiency of the rendezvous protocol does not compensate for the setup overhead until message sizes reach 32 KiBytes. The transition from eager to rendezvous protocol can be adjusted with the MVAPICH runtime parameter VIADEV_RENDEZVOUS_THRESHOLD**.**

The effect of the InfiniBand range extension upon latencies can be quantified by taking the difference between measured latencies in the IMB point-to-point results for small message sizes. For example, in the PingPong test for the HPC nodes, the measured latency for 1 Byte messages when running over the local InfiniBand fabric with no InfiniBand range extension was 4.0 microseconds. The same test when run over the Longbows connected with a 3 m single mode fiber patch cable was 9.9 microseconds. Since the latency induced by the speed of light in the fiber over 3m is negligible, we can infer that the inherent latency cost of InfiniBand range extension via a pair of Longbows is approximately 5.9 microseconds. By adding this inherent latency to the expected latency from the 690 m fiber link used to benchmark the HCS node configurations, we predict that the additional latency due to InfiniBand range extension for 1 Byte messages should be approximately 8.2 microseconds – we measure 8.0.

Further, from the "inherent" latency cost of 5.9 microseconds per Longbow link with negligible contribution from the speed of light, we can deduce that the latency through a single Longbow from the InfiniBand port to the optical port is approximately 1.48 microseconds. Not surprisingly, this latency is significantly larger than port-to-port latencies typical of 4X SDR InfiniBand switches, which typically run from 140-420 nanoseconds – normal InfiniBand switches do not have to do the work of encapsulating InfiniBand traffic into WAN protocols. Finally we note, as one might expect, that latency values are influenced by and are inversely proportional to CPU speeds down to the lower limit imposed by the InfiniBand interconnect itself.

The effect of the additional latency upon MPI throughput benchmarks is apparent, and not surprising. Throughput also varies (slightly) with CPU speed until the rate is capped by 1) the PCI-X bus bandwidth in the case of the HCS nodes or 2) the InfiniBand interconnect itself (10 Gb/s) in the case of the HPC nodes. Though the throughput achieved through the Longbows for a given message size was less than that achieved without the Longbows, the maximum throughputs observed were identical with and without the Longbows, in all configurations, and in all MPI tests. In the unidirectional PingPong test with the HPC nodes, the throughput achieved was over 960 MBytes/sec, with and without the Longbows – over 96% of the theoretical limit. The full-duplex nature of both the PCI-E cross-connect "bus" and InfiniBand interconnect can be seen in the IMB SendRecv and Exchange tests as bidirectional bandwidths reach levels in excess of 1900 MB/s (over 95% of the theoretical maximum). Similar wire-speed efficiencies were observed in the other MPI throughput results.

C. IOzone

Figure 16 shows single-threaded IOzone random write throughput for the HCS nodes in both CPU/memory configurations, with and without InfiniBand range-extension. We can see that as the block size grows larger than 256KiB, the throughput achieved with the HCS(240) node begins to plateau, ultimately leveling off at about 150 MBytes/sec. The throughput of the HCS(248) node, on the other hand, continues to increase until achieving a maximum of approximately 240 MBytes/sec. The ratio of these maximum throughputs is equal to the ratio of clock speeds of the CPUs. In both configurations, the throughput over Longbow-connected links is the same as the throughput achieved over the conventional InfiniBand fabric. The two-threaded aggregate random write throughput results shown in figure 17 show the same behavior with respect to the Longbows and the

same throughputs are achieved whether or not the iSCSI traffic is transported over the Longbows. The peak aggregate throughputs recorded were approximately 300 MBytes/sec for the HCS(240) nodes, and 450 MBytes/sec for the HCS(248) nodes. Again, the ratios of these maximum aggregate throughputs are roughly equal to the ratio of processor clock speeds. The additional latency introduced by the Longbows and the fiber length has no observable impact on random write throughput in these tests.

Figure 18 shows aggregate random write throughput for eight IOzone child processes running over the Longbows – two processes per HCS node (one per CPU). For the HCS(240) nodes, eight processes were not enough to saturate the link between the HCS nodes and the storage. The peak aggregate throughput observed was 790 MBytes/sec. The HCS(248) nodes were able to saturate the link and sustained an aggregate bandwidth of over 960 MBytes/sec (96% of the theoretical maximum).

In figure 19, aggregate random write throughput as a function of block size is measured for a single HPC node to the RapidScale (zXFS) file system atop the RAID level 0 LUNs described earlier. The results for each of the Longbow configurations are similar with aggregate throughputs in excess of 520 MBytes/sec for block sizes greater than 256 KiB. Again, the additional latency introduced by the Longbows does not have a significant effect upon the aggregate throughput.

Finally, figure 20 is similar to figure 18. In both cases, the storage is fast enough to handle the full bandwidth of an InfiniBand cable. The peak aggregate throughput was approximately 770 MBytes/sec for the HCS(240) nodes, and just over 960 MBytes/sec for the HCS(248) nodes.

V. Conclusion

In this paper, the Longbow InfiniBand range-extension devices from Obsidian Research Corporation have been characterized using several techniques and protocols: TTCP over SDP/IB, MPI over IB/VAPI, and iSCSI over SDP/IB (using the RapidScale parallel file system). For all three types of traffic, the Longbows demonstrated that they were capable of high wirespeed efficiency: over 97% of theoretical throughput was achieved with TTCP/SDP, over 96% for MPI, and over 96% for iSCSI.

In the storage arena, which is of particular interest to the HPC and Campus Grid efforts at the University of Florida, over 960 MBytes/sec of aggregate random write throughput was observed to the HPC storage system from four iSCSI initiators located at the HCS Lab, 690 m of fiber away – this shows that InfiniBand range-extension via Longbows can be a high throughput platform from which geographically distributed storage devices can be joined together, for example, into a parallel file system providing high performance storage access to an entire campus. While the effects of latency are most observable in latency-sensitive communications protocols like MPI, we found that latencies typical in a fiber network serving a campus like the University of Florida do not impact iSCSI throughput significantly.

Acknowledgment

The authors would like to thank Obsidian Longbow Limited Partnership for providing the Longbow units and fiber spool used in this paper. The authors would specifically like to acknowledge David Southwell, Jason Gunthorpe, and Bill Halina of Obsidian for useful discussions and support.

The authors would also like to acknowledge Gautham Sastri and Michael DeClerk of Rackable Systems for discussions and support regarding the RapidScale parallel file system, and Cisco Systems for their support regarding the HPC Center's InfiniBand fabric.

Further, the authors acknowledge the University of Florida High-Performance Computing and Simulation Research Lab and the University of Florida High-Performance Computing Committee for providing computational resources and support that have contributed to the research results reported within this paper. In particular, the authors would like to thank Dr. Alan George and Kyu Park for their support. Finally, the authors would like to acknowledge Dr. Alan George, Dr. Erik Deumens, Dr. Paul Avery, and Kyu Park for reviewing this paper in its draft form.

References

[1] Moore, "*Cramming more Components onto Integrated Circuits*", Electronics Magazine, April 1965.

[2] T. Noyes and W. E. Dickinson, "*The Random-Access Memory Accounting Machine - II. The Magnetic-Disk, Random-Access Memory*", IBM J. Res. Develop. 1, 72-75 (1957).

[3] David A. Patterson, Garth Gibson, and Randy H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," *Proceedings of the ACM SGIMOD International Conference on Management of Data*, pp.109-116, June 1988.

[4] C. Clos, "A study of non-blocking switching networks," Bell System Technical Journal, Vol. 32, 1953, pp. 406-424.

[5] Iain B. Findleton, "A New Paradigm for Parallel I/O to Resilient Network Storage", RapidScale Technologies, Inc., January 2006.

[6] W. Norcott, D. Capps, IOzone file system benchmark, http://www.iozone.org.

[7] S. Carter, M. Minich, and N. Rao, "Experimental evaluation of Infiniband Transport over local- and wide-area networks", *High Performance Computing Conference*, Norfolk, VA, Mar. 2007.

[8] M. Muuss and T. Slattery, http://ftp.arl.mil/ftp/pub/ttcp

[9] Intel MPI Benchmarks (Version 2.3), From the Intel Cluster Toolkit for Linux, http://www.intel.com.

[10] http://www.timbercon.com/Network-Simulations/index.html

[11] "MVAPICH: MPI for InfiniBand on VAPI Layer", Network-Based Computer Lab., The Ohio State University, http://nowlab.cis.ohio-state.edu/projects/mpi-iba/index.html.

[12] D. Goldenberg, M. Kagan, R. Ravid and S. Tsirkin, "Zero Copy Sockets Direct Protocol over InfiniBand – Preliminary Implementation and Performance Analysis," in *Proc. 13th Symposium on High Performance Interconnects,* Washington, D.C*.,* 2005, pp. 128-137.

[13] P. Balaji, S. Narravula, K. Vaidyanathan, S. Krishnamoorthy, J. Wu, and D. K. Panda, "Sockets Direct Protocol over InfiniBand in Clusters: Is it Beneficial?", in proc. Of IEEE International Symposium on Performance Analysis of Systems and Software ((ISPASS 04), Mar. 2004, pp. 28-35.

[14] H. Tezuka, F. O'Carroll, A. Hori, and Yutaka Ishikawa, "Pin-down Cache: A Virtual Memory Management Technique for Zero-copy Communication", in Proc. of the 12th International Parallel Processing Symposium, Mar. 1998, pp. 308-314.

[15] J. Liu, A. Mamidala, A. Vishnu, D.K. Panda., "Performance Evaluation of InfiniBand with PCI Express"*, IEEE Micro*, Vol. 25, Issue 1, Jan-Feb, 2005, pp. 20-29.

[16] Mellanox Technologies, "Mellanox InfiniBand InfiniHost III EX MT25208 Adapters", http://www.mellanox.com, Feb. 2004.

[17] The OpenFabrics Alliance, http://www.openfabrics.org.

[18] D. Pase, "Performance of Voltaire InfiniBand in IBM 64-Bit COmmodity HPC Clusters", Available : ftp://ftp.software.ibm.com/eserver/benchmarks/wp_IB_Performance_053105.pdf