

Evaluation of an Integrated PCI Express IO Expansion and Clustering Fabric

Venkata Krishnan

Dolphin Interconnect Solutions

Marlborough, MA 01752

krishnan@dolphinics.com

Abstract— Dolphin’s DX interconnect, comprising of PCI Express based hardware and accompanying software is an industry first solution for seamlessly integrating IO and clustering capabilities onto an enhanced PCI Express interconnect. This one of a kind solution eliminates the need for two distinct interconnects – one dedicated to IO and the other to clustering.

The DX components can enable IO expansion in a single-host platform and provide clustering as well as IO expansion support in a multi-host platform. The hardware components of the DX solution consist of a (i) host adapter (ii) expansion switch and (iii) cluster switch. The software components provide the critical infrastructure for allowing legacy applications to run without the need for code rewrite or recompilation. More specifically, the sockets direct interface in place (termed as *SuperSockets*) allows the TCP/IP stack to be completely bypassed and can be used in a transparent manner to enhance the performance of legacy socket applications.

This paper evaluates the performance of an integrated clustering and single host IO expansion platform using DX components. Results show that the DX components add minimal overhead to bandwidth and latency from a single host IO expansion perspective. From a clustering perspective, preliminary results using the SuperSockets approach show a 1-byte packet latency of $\sim 2\mu\text{s}$ for TCP/IP/socket based applications – an order of magnitude lower than what is currently available. The paper also evaluates a prototype gateway model that builds upon the combined single-host IO expansion and clustering feature thereby obviating the need for additional bridging components. Such a gateway model permits hosts in the cluster to have seamless connectivity to external networks through centralized network IO adapters.

I. INTRODUCTION

PCI Express (PCIe), the interconnect of choice in the IO and chip-to-chip communication realm, will include support for IO virtualization both in a single-host and multi-host environment[1]. Nonetheless, host-to-host communication is considered outside the scope of PCIe. Dolphin’s DX solution addresses this shortcoming in PCIe.

The DX solution, comprising of PCIe-based hardware and accompanying software, is an industry first for seamlessly integrating IO and clustering capabilities onto an enhanced PCIe interconnect thereby eliminating the need for two distinct interconnect domains – one dedicated to IO expansion and the other for clustering. Furthermore, the solution is not

only transparent to legacy PCIe software but can also support traditional cluster applications using standard APIs (e.g. TCP/IP/sockets). Hence, there is no need for code modification or recompilation to take full advantage of this platform.

IO expansion capability permits isolation of the IO subsystem from the main processor complex and offers key advantages on several fronts. On the high availability front, when a primary IO device fails, a redundant IO device can take over without the need for bringing down the processor sub-system. On the resource allocation front, IO expansion offers an adaptive infrastructure in that resources can be apportioned to different processor sub-systems according to their needs; finally, on the system upgrade front, upgrades become cost effective for the simple reason that the IO and processor subsystems can be treated as independent entities. To facilitate the perception of a single logical system, a switched interconnect is needed to connect the IO and processor sub-system domains.

Potentially, the same interconnect that permits PCIe IO expansion capabilities can also serve as a clustering interconnect and support host-to-host communication. Though seemingly elegant, supporting such a dual functionality is not trivial. The underlying hardware components need to be augmented along the lines of a clustering interconnect such as Ethernet or Infiniband[2]. In addition, the software components must support legacy APIs – specifically the sockets API. Indeed, in the software world of cluster communication, sockets API or its variants have been the de-facto interface for a wide body of networking applications. The level of entrenchment of sockets-based applications is so deep that it is essential, for a clustering interconnect technology, to provide a completely transparent socket interface - one that does not require existing applications to be modified or recompiled.

This paper briefly describes the various DX components that enable such an integrated clustering and IO expansion solution. This includes the hardware components that consist of a host adapter, an expansion switch and a cluster switch. Furthermore, the paper explains how the IO expansion capability can be provided to the host using the host adapter and expansion switch. The software infrastructure that enables legacy cluster applications to take full advantage of the underlying interconnect is also described.

A significant portion of the paper is devoted to the performance evaluation of a platform that simultaneously

supports single-host IO expansion as well as host-to-host communication. In addition, a prototype implementation of a gateway model is also evaluated. In this gateway model, the IO expansion capability and the clustering components work in tandem and enable the hosts to have seamless connectivity to an external network without requiring additional bridges.

The rest of the paper is organized as follows: Section II describes the DX hardware components while Section III gives a brief overview of the software infrastructure. Section IV evaluates the single-host IO expansion and clustering platform; finally Section V concludes the paper.

II. DX HARDWARE

Dolphin's DX hardware brings new functionality to standard PCIe by enabling (i) IO expansion in a single-host platform and (ii) IO expansion as well as clustering in a multi-host platform. The hardware components required to support these two solutions consists of a PCIe host adapter, expansion switch and a cluster switch.

A. DXH510: Host Adapter

Dolphin's DXH510 PCIe host adapter (see Figure 1) provides the necessary functionality to add clustering and IO Expansion functionality to standard PCIe.

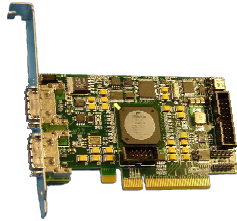


Figure 1. DXH510: A low profile form factor PCI Express Host Adapter that supports two x4 PCI Express high speed output connectors and uses an x8 PCI Express interface to the host subsystem.

On the IO expansion front, the host adapter (HA) encapsulates PCIe packets for transport which is then extracted by the expansion switch (described in a section II-B). There is *no need* for any additional bridging device between the PCIe IO device and an expansion switch slot. Furthermore, the connection establishment between the host and IO device is done transparently without any additional software. Indeed, all data movement between the host and IO device continue to be under the control of legacy PCI/PCIe software.

On the clustering front, the HA incorporates protocols that permit host-to-host communication. The host-to-host data movement protocol has two modes of transfer. The first mode of transfer uses a *send/receive* model wherein messages are transmitted to a queue at the receiving host system. The queues provide a secure, in-order mechanism for delivering messages from multiple endpoints. The HA supports multiple such queues – these queues can be used to support a connection-less datagram model. Alternatively, a given queue can be configured such that access to it can be limited to a (i)

single host (ii) set of hosts (iii) particular process in a specific host/set of hosts. Messages are written into the queues, which are ring-based structures in the local memory of the host subsystem. Along with the payload, these messages contain additional information for post-processing by software. The HA generates messages for the target queue using its DMA engines.

The second mode of data transfer is memory based – the HA supports mechanisms for accessing the system memory of a remote host. The need for exposing the physical memory for remote access is avoided by the use of security tags and more importantly, a memory indirection mechanism. Thus, any remote memory access is screened and validated by the HA before being directed to system memory. Finally, the HA provides the flexibility for remote memory to be accessed using either the programmed IO (PIO) or Remote Direct Memory Access (RDMA) paradigm.

The DXH510 host adapter supports two industry standard x4 PCIe high speed connectors. Each of the two ports supported by DXH510 can be connected to (a) another DSXH510 host adapter (b) cluster/IO expansion switch or (c) a cluster switch. The ports support either copper or fiber optic cabling with the use of pluggable transceivers. Each port is capable of up to 10Gb/s transfer rates and can also be combined to support 20Gb/s. Indeed, at a bandwidth of 20Gb/s, the DXH510 can satisfy the needs of most applications.

B. DXE410: Expansion Switch

The DXE410 4RU Expansion Switch (Figure 2) is unique for its flexibility, ease of use, and scalability. For IO Expansion, the DXE410 adds both PCIe slot and distance scalability to low cost servers and workstations with limited PCIe slots. Enterprise and Embedded application developers can lower system cost and cooling requirements by disaggregating their IO from servers and workstations. The DXE410 supports both copper and fiber optic cabling, allowing it to connect to a host up to 100 meters away. In conjunction with the DXH510 PCIe Host Adapter, transfer rates between a host and the expansion chassis can reach up to 20Gb/s.

As shown in Figure 2, the DXE410 supports both x4 and x8 PCIe slots. A total of eight configurable PCIe slots are available. The chassis is self configuring based on which slots are selected in the chassis, so multiple configurations are available. Applications requiring higher performance can configure the chassis for up to four x8 PCIe slots, while applications requiring more slots can be configured for eight x4 PCIe slots. Not all slots may be used in an x8 mode (see Figure 2). Furthermore, if a slot is to be used in x8 mode, the adjacent x4 slot on the right must be left unused.

Dolphin implemented the DXE410 with several ease of use features, such as self configuring slots and remote power-on via fiber optics. The remote power-on feature provides the capability to locate the expansion chassis up to 100 meters away from the host, but the DXE410 will power-on when the host powers on. Remote power-on aids in system deployment and maintenance, reducing setup time and configuration. The

DXE410 is a flexible system that allows users to scale both distance and performance.

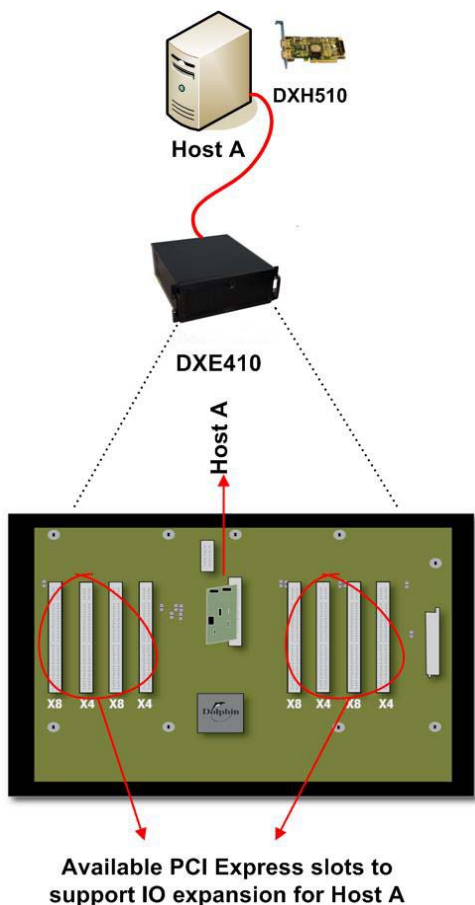


Figure 2. DXE410: A 4RU Cluster/IO Expansion Switch. The chassis supports up to eight downstream x4 PCI Express slots and an x8 uplink.

In addition to using the PCIe slots for IO expansion, the slots can also be potentially used for clustering. This can be achieved in a straightforward manner by using a break-out or riser board on the slot which in turn is connected to a host’s DXH510 adapter. For instance, if the cluster consists of only four hosts and requires only four IO devices to be supported, a single DXE410 would be sufficient.

C. DXS410: Cluster Switch

The DXE410 expansion switch may suffice if the clustering and IO expansion requirements are limited. However, for building larger clusters, additional switches may be required. Since such switches require only the capability to provide clustering, Dolphin’s hardware components include the DXS410 – a 1RU cluster switch (See Figure 3) that supports up to a total of ten configurable PCIe based x4 ports. Based on the requirements of the application, a pair of x4 ports can be configured to one x8 port.

A single DXS410 can be used for creating up to a 10 node cluster. Each port on the DXS410 is connected directly to the DXH510 resident in 10 different servers. A second DXS410

can be added to create a redundant connection between each of the 10 nodes to provide high availability benefits. In this case one of the ports on the DXH510 is connected to one of the switches and the remaining port is connected to the other switch. In case of a link or switch failure, traffic can be re-routed through the redundant fabric. Multiple DXS410s can also be connected in various configurations to support larger and larger cluster sizes, efficiently supporting clusters up to 64 nodes and beyond.



Figure 3. DXS410: A 1RU Cluster Switch which supports up to (a) ten x4 ports or (b) five x8 ports or (c) a combination of x4 and x8 ports.

Using the three DX hardware components: the HA, the cluster switch, and the expansion switch a rich variety of cluster and IO configurations are available supporting the tailoring of the appropriate mix of elements to optimize the performance of any particular application.

III. SOFTWARE

Dolphin’s DX software supports both legacy PCI/PCIe and clustering. In fact, no software is required for supporting IO expansion in a single-host platform. On a multi-host platform, a configuration tool will be available to enable flexible (re)assignment of a PCIe IO device to an arbitrary host.

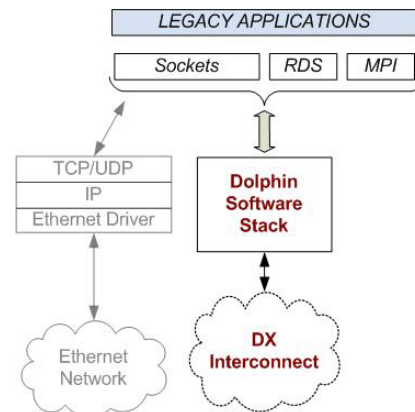


Figure 4. DX clustering software will continue to support legacy applications using standard APIs. The sockets direct approach (SuperSockets) completely bypasses the TCP/IP stack and enables ultra-low latency and high bandwidth communication.

Sockets API is generally considered the de-facto interface in the world of networking applications. DX clustering

software (Figure 4) supports various application interfaces and in particular, the sockets API. A sockets direct approach (termed as *SuperSockets*) that bypasses the TCP/IP stack enables ultra low latency and high bandwidth to be achieved.

IV. EVALUATION

This section focuses on the performance of the integrated model that supports clustering as well as single host IO expansion (See Figure 5). Sample applications of this model include a desktop supercomputing cluster which permits high performance computing (using accelerator cards dedicated to a primary host) and high speed communication amongst the hosts using the DX interconnect; small database clusters wherein only the primary node communicates with a SAN via a FC adapter located in the expansion switch box; medical imaging/rendering where in a single host is responsible for data acquisition etc.

Given that the IO expansion and clustering performance are rather orthogonal, the evaluation is split into two components – one for the single host IO expansion and one for the clustering.

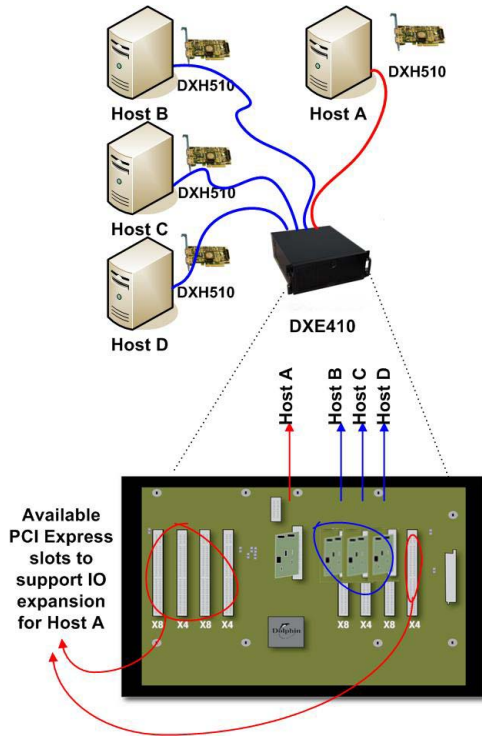


Figure 5. An integrated platform that simultaneously supports PCIe based IO expansion for a single host and clustering.

D. Single Host IO Expansion Performance

To determine the performance impact of using the expansion switch, a two host 10GigE and Infiniband cluster was used wherein the hosts were connected point-to-point without any intermediary switch.

The 10GigE adapters used for the setup were Myri-10G[3] with an x8 PCIe interface. To establish a baseline performance,

the 10GigE adapters were attached directly to the host motherboard's x8 PCI express slots. Such a direct attached configuration is shown in Figure 6(a).

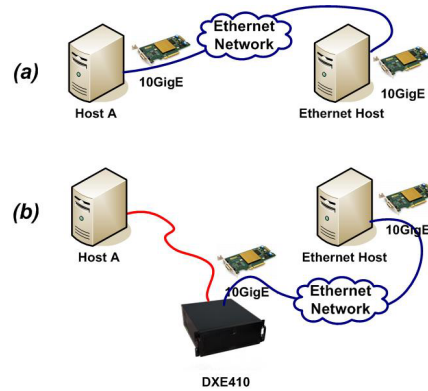


Figure 6. Illustration describes the disaggregation of a 10GigE NIC. The location of the NIC for host A can be (a) Direct (b) Expansion switch based.

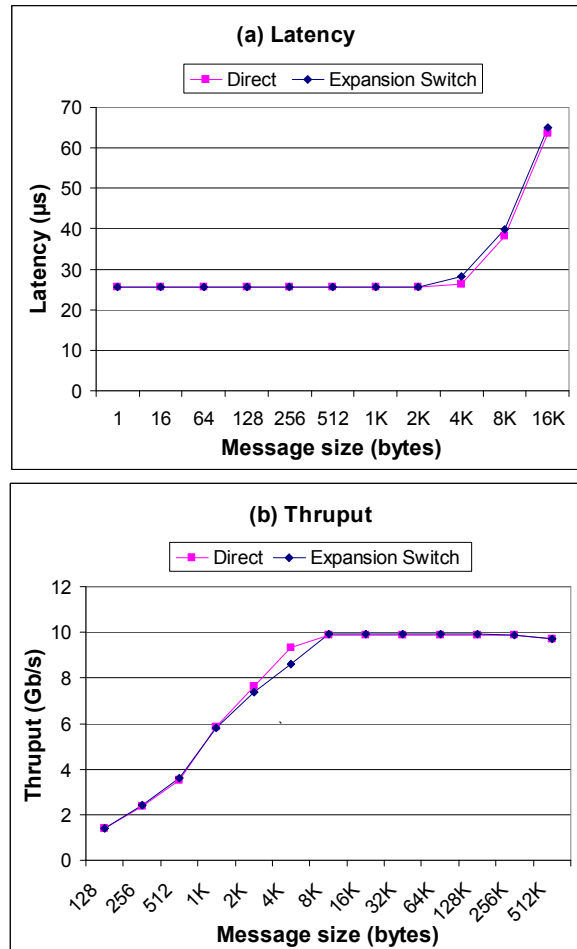


Figure 7. Performance impact of the IO expansion switch on a 10GigE NIC.

For the second setup, a 10GigE adapter was decoupled from one of the hosts and moved to the expansion switch box. Figure 6(b) illustrates this configuration. Recall from Figure 5 that there are two additional devices on the 10GigE data path – the host adapter (DXH510) in Host A and the expansion switch (DXE410).

Figure 7(a) and (b) chart the latency and throughput respectively for the two configurations. The measurements were obtained using the standard TCP/IP networking benchmark, Netperf. The data shows an imperceptible change in the latency or throughput for the expansion switch based configuration. In other words, the inclusion of two DX devices on the 10GigE data path has no impact on the latencies of small messages (below 4KB) while increasing the latencies of large messages (over 4KB) by at most 1.5 μ s. There is also little or no change from a bandwidth perspective - the performance drop ranges from between 1% to 5%.

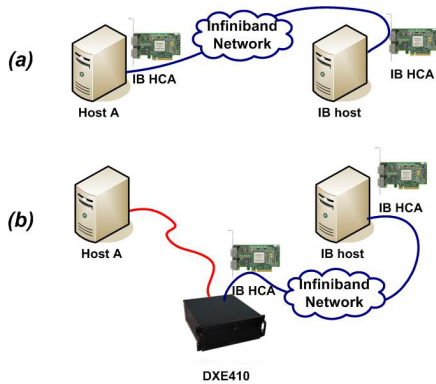


Figure 8. Illustration describes the disaggregation of an Infiniband HCA. The location of the HCA for host A can be (a) Direct (b) Expansion switch based.

Moving on to the infiniband configuration, the setup is quite similar to that of the 10GigE's. The HCAs used were Mellanox InfiniHost III Ex [5] with an x8 PCIe interface. As before, the adapters were attached directly to the host motherboard's x8 PCI express slots and also via the expansion switch – see Figure 8(a) and (b). The best case latency and bandwidth measurements were obtained using the performance tests that are part of the OFED distribution[6].

Figure 9(a) and (b) show the performance impact of the expansion switch on the HCA's latency and throughput. Unlike the 10GigE configuration, the IB configuration exhibits very low latencies for a wide range of packet sizes. For instance, the 1-byte message latency for the 10GigE configuration is 25 μ s as opposed to 2.6 μ s for the IB configuration. Nonetheless, even in a configuration in which latency is extremely small, the addition of the DX components do not result in significant performance degradation. The data shows a marginal increase of 0.7 μ s for a 1-byte message and up to 1.5 μ s for large messages (over 1K). Additionally, there is also little or no impact on the bandwidth.

E. Host to Host Communication

A distinguishing feature of the DX interconnect is its support for clustering within the same PCIe expansion fabric.

From a clustering software perspective, a first step towards supporting a comprehensive clustering software solution was the development of an IP driver. This driver facilitated the support of TCP/IP/sockets over PCIe (IPoPCIe). For additional details, please see [7].

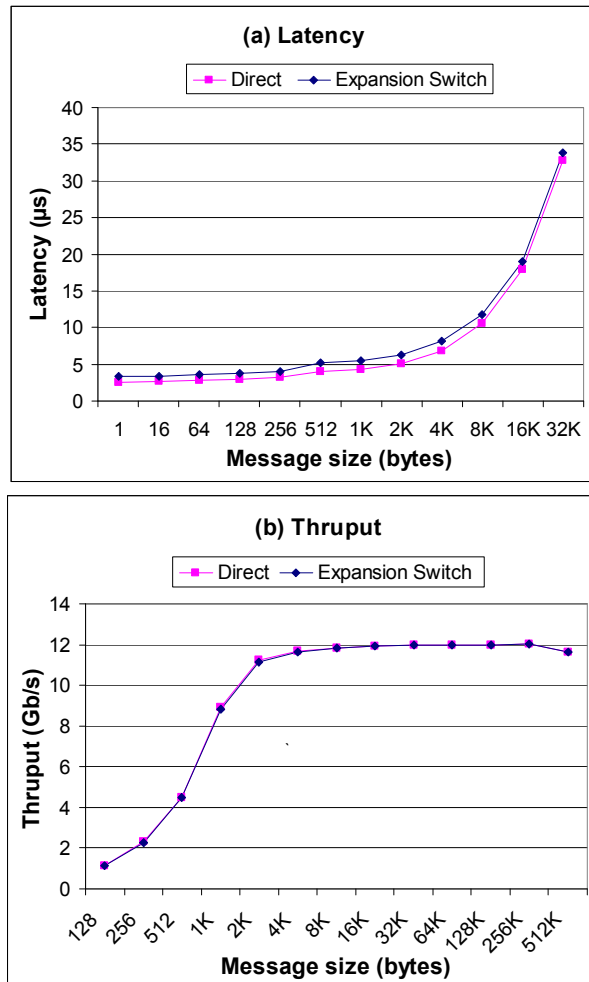


Figure 9. Performance impact of the IO expansion switch on an Infiniband HCA. The best case latency and throughput numbers shown above were obtained using the OFED performance tests (rather than the standard sockets API based Netperf)

Given the lossless nature of the DX fabric as well as other capabilities afforded by the DX host adapter (See Section II), the use of the complete TCP/IP stack is rather unnecessary. Indeed, Dolphin supports a direct sockets interface that bypasses the TCP/IP stack. This interface, termed as SuperSockets[8] permits ultra-low latency and higher bandwidth to be achieved. For the subsequent experiments, the DX interconnect is treated as a traditional clustering interconnect wherein the hosts communicate with each other. This is shown in Figure 10. The latency and throughput for SuperSockets were measured using the sockets based Netperf benchmark. It should be noted that the benchmark was neither modified nor recompiled for the measurements. Rather, the user level socket calls were intercepted at run time with a

dynamically linked library. This library, in turn, was responsible for steering the calls to the SuperSockets interface or to the native TCP/IP sockets interface.

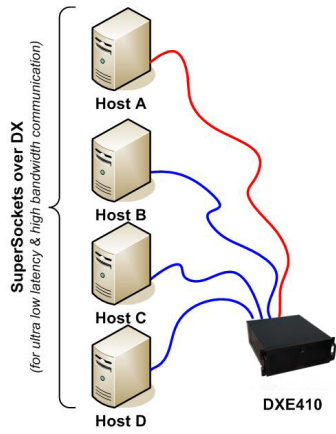


Figure 10. Using DX interconnect for clustering.

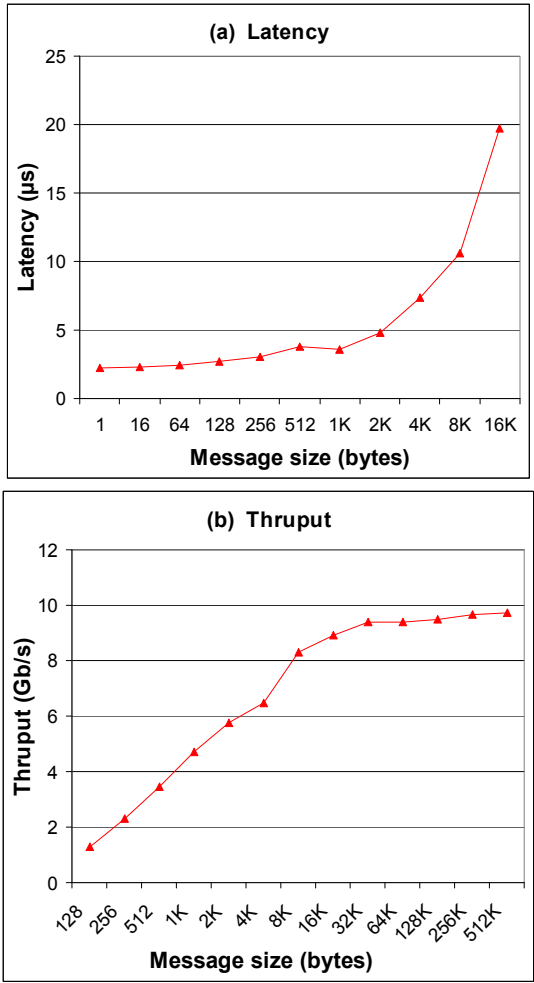


Figure 11. Netperf latency and bandwidth measurements on a DX clustering interconnect using the TCP/IP bypass sockets interface (SuperSockets).

Figure 11(a) plots the latencies for various message sizes. The SuperSockets approach results in a one-way latency of ~2μs for a 1-byte message. To the best of our knowledge, this is the lowest latency that has been achieved for legacy socket applications. Such a low latency is facilitated by the PCIe-like characteristics of DX interconnect.

As with other interconnects such as Ethernet or Infiniband, DX supports a packet-based paradigm wherein DMA is used for data transfers both at the source and the receive side. At the same time, for very small messages, the overhead of using DMA far outweighs potential benefits. Even though some optimizations[9] may be performed to reduce this impact, such an approach may not be applicable for a vast pool of legacy socket applications wherein the source code can neither be modified nor recompiled. Accordingly, the DX solution uses a different approach.

DX supports a programmed IO (PIO) based packet generation model[10][11] and in addition permits accesses not only to the memory of IO devices but also to that of remote hosts. And unlike some of the non-transparent PCIe bridge approaches that directly expose remote memory[12], DX utilizes a tagged memory access mechanism that results in a very secure model for remote data accesses. This is along the lines of RDMA enabled NICs[13]. The SuperSockets software, in turn, exploits these features in an optimal manner and chooses the appropriate mode (either PIO or DMA) for transmitting a packet of a given size. Hence the ultra-low latencies for legacy socket applications.

Figure 11(b) charts the throughput for varying packet sizes using the SuperSockets interface. Even though the observed performance is notable, there is still scope for improvement. At the time of writing this paper, support for DMA by SuperSockets software was still work under progress. The bandwidth numbers are expected to improve once the DMA component is in place.

F. Gateway Model

The earlier sections treated single host IO and clustering as separate entities even though both were supported simultaneously on the same platform. One potential usage model that arises out of a combined configuration is the support for a gateway model. In this scenario, the primary IO expansion host can be used as a gateway and enable other DX hosts to have seamless access to the external network. Dedicated router/gateway hardware is currently available for interconnects such as Infiniband for bridging IB-to-IP[14] and IB-to-FC[15] networks. In contrast, the approach proposed here is unique in that it builds upon the expansion switch capability without requiring additional bridging hardware components. Such an approach also allows consolidation as well as co-location of primary/failover PCIe-based IO devices (such as 10GigE or FC adapters) away from the primary host and in the process enable low-form factor hosts to be used. As described in Section I, this is a key criterion towards building cost-effective servers.

As a proof of concept, the IO expansion host was chosen as a TCP/IP forwarding gateway. This is shown in Figure 12(a). All the hosts use the IP driver (IPoPCIe[7]) for forwarding

TCP/IP packets to the gateway. The evaluation is done along the lines of single host IO expansion. In the baseline setup shown in Figure 12(a), the 10GigE adapter resides in the expansion host/gateway (Host A) and is attached directly to the Ethernet host. In Figure 12(b), the 10GigE adapter is migrated to the expansion switch. With this change, the connectivity to an external 10GigE network for all the hosts is *centralized* at the expansion switch.

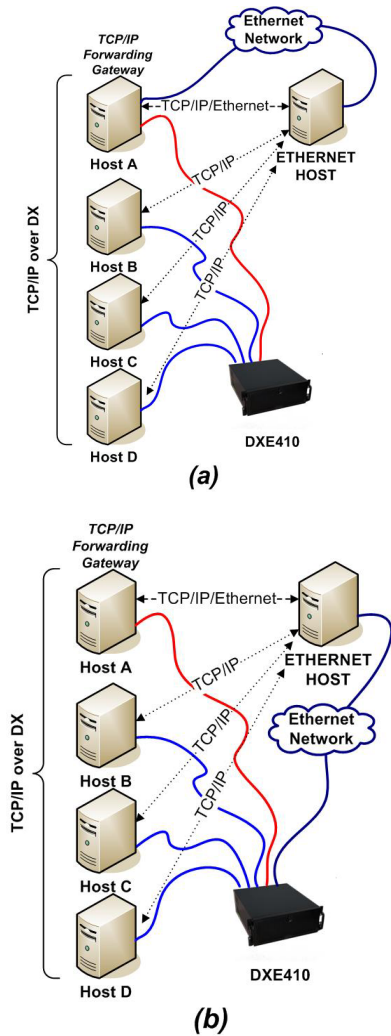


Figure 12. Supporting a gateway model using the DX interconnect.

For the first set of experiments, a gateway fan-in of 3 processes was used wherein hosts B, C and D communicated with the external Ethernet host (via the gateway host). Figure 13(a) and (b) compare the latencies when used in a direct attached as well as in an expansion configuration. As in the single-host IO expansion case, there is little impact on latency and all three processes have nearly comparable values for different packet sizes.

As with the latency measurements, the throughput measurements for the fan-in mode show little or no variation

between the direct attached and expansion mode - see Figure 14 (a) and (b). Given that the throughput does not vary wildly across different processes, it is apparent that the processes are being serviced in a fair manner. However, due to forwarding overheads, the cumulative throughput available for the hosts is 7Gb/s (a drop of ~2.5Gb/s)

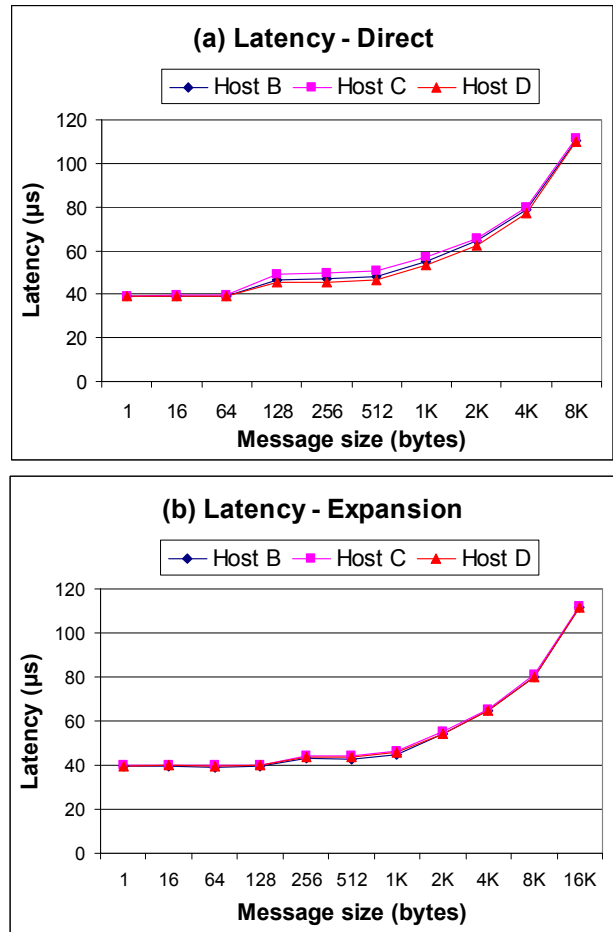


Figure 13. Latencies for a gateway fan-in of 3 processes in (a) Direct attached and (b) Expansion switch based configuration.

V. CONCLUSIONS & FUTURE WORK

Dolphin's DX hardware and software components provide the necessary wherewithal to address some of the shortcomings of PCIe – namely in the areas of IO expansion over long distances and support for host-to-host communication. The DX solution doesn't require changes at the application level and indeed, legacy applications can exploit the full potential of the underlying enhanced PCIe interconnect without the need for code rewrite or recompilation.

The underlying DX components offer a rich variety of configurations – currently, a single host IO expansion and clustering platform is supported. Results based on this configuration show that the DX components add little overhead in a single host IO expansion scenario while

providing ultra-low latencies and high bandwidth in a clustering setup. The integrated platform also enables a unique gateway model that obviates the need for additional bridging hardware components and yet at the same time, provides a centralized model for accessing network IO adapters.

work includes supporting a wide variety of DX configurations that include a very flexible way of assigning an arbitrary IO device to an arbitrary host.

VI. ACKNOWLEDGEMENTS

The author is grateful to the Dolphin engineering team and in particular, is indebted to the following people who made this work possible - Lynne Brocco, Todd Comins, Hugo Kohmann, Roy Nordstrom, Friedrich Seifert, Tom Tinory, Mike Vasicek, Atle Vesterkjaer, David Wong and Joachim Worringen.

REFERENCES

- [1] PCI Special Interest Group, "PCI Express 2.0 Base Specification," January 2007.
- [2] Infiniband Trade Association, "Infiniband Architecture Specification, Release 1.2," Sep 2004. <http://www.infinibandta.org>
- [3] Myri-10G PCI Express NIC with a 10GBase-CX4 port, Myricom Inc. <http://www.myricom.com/Myri-10G/NIC/10G-PCIE-8A-C.html>
- [4] Netperf: A Network Performance Benchmark Revision 2.4.1, October 2005. <http://www.netperf.org>.
- [5] InfiniHost™ III Ex Dual-Port InfiniBand HCA Cards, Mellanox Technologies. <http://www.mellanox.com/pdf/products/hca/IH3EX.pdf>
- [6] Open Fabrics Enterprise Distribution (OFED), Mellanox Technologies www.mellanox.com/pdf/products/software/OFED_PB_1.3.pdf
- [7] V. Krishnan, "Towards an Integrated IO and Clustering Solution using PCI Express", IEEE International Conference on Cluster Computing (CLUSTER 2007), September, 2007.
- [8] F. Seifert and H. Kohmann, "SCI Socket - A Fast Socket Implementation over SCI", Dolphin Interconnect Solutions. <http://www.dolphinics.com/userfiles/files/Whitepaper/sci-socket.pdf>
- [9] C. Bell, and D. Bonachea, "A new DMA registration strategy for pinning-based high performance networks", IEEE International Parallel & Distributed Processing Symposium (IPDPS), April 2003.
- [10] K. Watanabe et al, "Martini: A Network Interface Controller Chip for High Performance Computing with Distributed PCs", IEEE Transactions on Parallel and Distributed Systems, September 2007.
- [11] Andrew Gallatin, Jeff Chase, and Ken Yocum, "Trapeze/IP: TCP/IP at near Gigabit Speeds", USENIX 1999.
- [12] S. Cooper, "PCI Express Outside the Box", RTC December 2007.
- [13] A. Romanow. An Overview of RDMA over IP. In First International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2003), February 2003.
- [14] Voltaire IP Router Module., Voltaire Inc., http://www.voltaire.com/download/datasheets/Voltaire_IPRouterModule_web.pdf
- [15] Voltaire Fiber Channel Router Module, Voltaire Inc., http://www.voltaire.com/download/datasheets/FCR_WEB.pdf

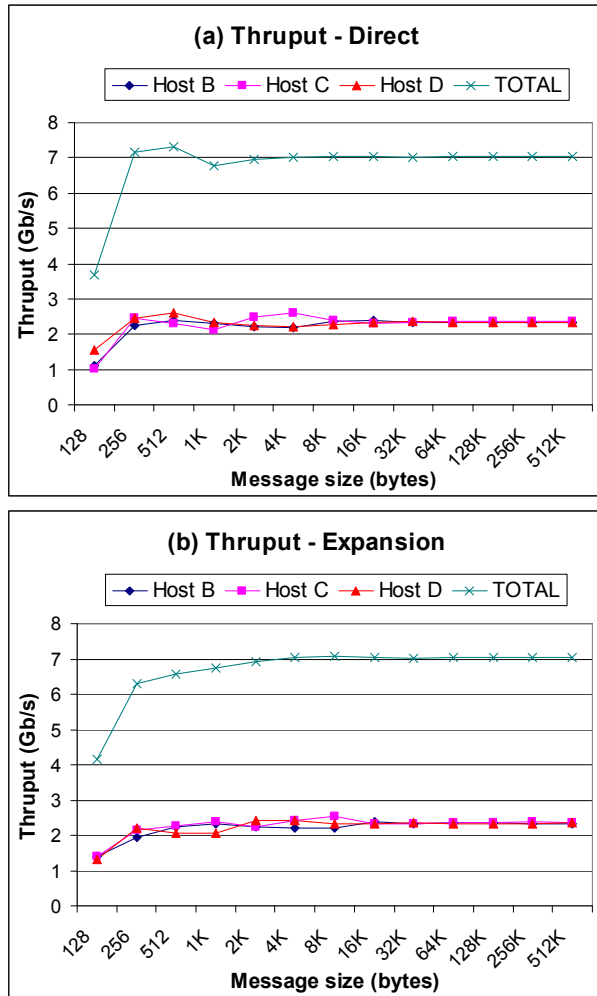


Figure 14. Throughput for a gateway fan-in of 3 processes in (a) Direct attached and (b) Expansion switch based configuration.

Current work is focused on completion/tuning of the SuperSockets driver from a throughput perspective. Future