

Evaluation of ConnectX Virtual Protocol Interconnect for Data Centers

Ryan E. Grant¹

Ahmad Afsahi¹

Pavan Balaji²

¹*Department of Electrical and Computer Engineering
Queen's University, Kingston, ON, Canada
{ryan.grant, ahmad.afsahi}@queensu.ca*

²*Mathematics and Computer Science Division
Argonne National Laboratory, Argonne, IL 60439, USA
balaji@mcs.anl.gov*

Abstract

With the emergence of new technologies such as Virtual Protocol Interconnect (VPI) for the modern data center, the separation between commodity networking technology and high-performance interconnects is shrinking. With VPI, a single network adapter on a data center server can easily be configured to use one port to interface with Ethernet traffic and another port to interface with high-bandwidth, low-latency InfiniBand technology. In this paper, we evaluate ConnectX VPI using microbenchmarks as well as real traces from a three-tier data center architecture. We find that with VPI each network segment in the data center can use the most optimal configuration (whether InfiniBand or Ethernet) without having to fall back to the lowest common denominator, as is currently the case. Our results show a maximum 26.7% increase in bandwidth, a 54.5% reduction in latency, and a 5% increase in real data center throughput.

1. Introduction

A large number of high-speed interconnects have been introduced in the market over the past few decades, including InfiniBand (IB) [9], 10-Gigabit Ethernet (10GigE) and its variants [17], Myrinet [13], and Quadrics [3]. While most of these interconnects provide similar functionality, they are incompatible with each other, thus imposing a usability bottleneck for many organizations. Since a single-step network upgrade is not realistic for many enterprise computing domains, they are forced into backward-compatible upgrades (e.g., 1GigE to 10GigE), which allow a small segment of the network to be upgraded while retaining the rest of the infrastructure. This approach, however, is fundamentally limited because once one networking technology is used to build the network backbone, it can no longer make use of enhancements from other networking technologies.

Arguably, some enhancements in networks such as Myrinet-10G and Mellanox ConnectX [12] allow the same network adapter to function either in a native mode (i.e., natively as Myrinet or InfiniBand) or in Ethernet mode. In Ethernet mode, such adapters can be seamlessly integrated into the existing network infrastructure. While this integration provides some benefit, it is not perfect because, as when configured in native mode, the adapter

loses Ethernet connectivity.

Virtual Protocol Interconnect (VPI) [4] is a recently introduced concept by Mellanox Technologies that allows an adapter to transparently migrate between native mode and Ethernet mode without requiring manual reconfiguration. With the introduction of VPI in ConnectX, we have for the first time a network fabric that can seamlessly and transparently bridge a high-speed networking technology with Ethernet. More specifically, VPI allows operating individual physical ports of a two-port ConnectX adapter in 10GigE or native IB mode. Thus, one can easily configure systems with one 10GigE port and one IB port using the same ConnectX adapter.

Such technology is especially important in enterprise data centers, for example, where a large local cluster of compute systems have to interact with each other (possibly using IB) as well as with external remote clients (with a backward compatible network such as Ethernet). Previously, high-speed networking technologies were impractical, since bridging these technologies with Ethernet required the use of special switches and incurred additional delay because of the additional switch layer. VPI, on the other hand, facilitates the use of hybrid interconnect architectures in such data centers, allowing the compute systems to interact in native IB mode, while allowing them to interact with the remote clients in Ethernet mode. VPI also allows for easier deployment of non-Ethernet network technologies in data centers by providing a seamless socket-based interface over IB. This provides for increased local bandwidth and can take advantage of advanced RDMA features and latency improvements, which are important in increasing the number of clients that application and database tiers can service, particularly for higher bandwidth content. In addition, VPI can provide space and cost benefits over using discrete adapters in an IB/Ethernet network configuration.

In this paper we explore the behavior of systems utilizing VPI, and we investigate the performance benefits that can be realized by allowing each network segment to operate in its most optimal configuration; that is, systems that can communicate with a 16 Gb/s IB DDR network configuration should do so, while systems that require network infrastructure compatibility can use a 10 Gb/s

Ethernet network configuration. We find that such flexibility reduces the intrasystem communication latencies of a 10 Gb/s Ethernet configuration by an average of 54.4% as compared to the base case. Similarly, in real data center environments where such configuration is critical, we notice that throughput can be increased by 5% and aggregate bandwidth can be increased to within 1.2% of the raw IB verbs bandwidth of our systems.

In Section 2 we provide a brief background on IB, ConnectX, and the associated communication protocols, IPoIB and Sockets Direct Protocol (SDP) [9]. Section 3 reviews related work. Section 4 introduces our experimental platform. In Section 5, we present baseline single- and multiple-stream results for 10GigE, SDP, and IPoIB traffic. We then evaluate VPI in terms of simultaneous bandwidth and in a real data center test, comparing the performance of different local area network backbones for an on-line bookstore benchmark. In Section 6 we draw conclusions and outline future work.

2. Background

InfiniBand is a leading high-performance networking technology. It utilizes a low-level *verbs* layer that forms the foundation of the InfiniBand software layer. Using InfiniBand verbs, one can access the network adapter directly, bypassing the local operating system. The verbs layer utilizes a queue pair model, providing support for both channel-based communication semantics (send/receive) and memory-based communication semantics (*Remote Direct Memory Access*, or RDMA).

The ConnectX InfiniBand network adapters are the latest generation of InfiniBand network cards available from Mellanox Technologies. ConnectX is a 4X InfiniBand two-port card capable of operating each independent port in either 10GigE mode or native InfiniBand mode. It provides a stateless offload capability for both Ethernet and InfiniBand protocols. In addition, ConnectX supports a number of extended InfiniBand features.

Communication in 10GigE mode goes through the traditional kernel-based TCP/IP stack but uses some network enhancements to provide stateless-offload (such as Large Segment Offload) capabilities. Communication in native IB mode can use either the verbs interface or high-level communication stacks. Example high-level communication stacks include the Message Passing Interface (MPI), the traditional kernel-based TCP/IP stack (through a driver that does Ethernet emulation, called as IPoIB), and high-level sockets emulation frameworks such as SDP. While MPI is the de facto standard programming model for scientific computing, sockets-based frameworks (such as IPoIB and SDP) are more prominent in enterprise data center environments. Thus, we consider only these two protocols in this paper.

IPoIB functions within the InfiniBand semantics by

encapsulating a full IP packet into an InfiniBand data packet. This provides the functionality of an IP-based network, while conforming to the requirements of the packet semantics in the InfiniBand standard.

SDP is a byte-stream-based protocol that takes advantage of the RDMA semantics available in InfiniBand to improve the performance of sockets. It advertises available buffers to potential peers when establishing a connection and can offer both a buffered-copy and zero-copy based transfer mechanism. Because of the costs of memory registration and connection establishment, zero-copy is more useful for larger message sizes than for smaller ones. Therefore, zero-copy and buffered-copy methods are both offered in the software stack, with a message size threshold switching point at which zero-copying is allowed to occur.

3. Related Work

Shah et al. [18] were the first to propose a high-performance socket for the VIA architecture. In previous work [2] we examined the performance of SDP and IPoIB in a data center context and found that both can have significant positive impact on the performance of data centers. However, such work utilized an InfiniBand-based network with no Ethernet functionality.

Goldenberg et al. [5] examined the potential performance improvement of adding zero-copy SDP functionality to the existing SDP implementation. The authors found that improvements could be made to throughput for larger message sizes. We subsequently introduced an asynchronous zero-copy mechanism [1] providing better performance than that obtained by Goldenberg et al.

Zhang et al. [22] investigated the performance of InfiniBand socket-based protocols in relation to the Java communication stack. They found that the performance of the existing Java communication stack is poor and requires changes in order to provide performance inline with the capabilities of the network fabric. Other work on the Java communication stack was done by Huang et al. [8], who added RDMA operation functionality for Java through a specialized library. The work to date with Java communication stacks differs from our uses of Java in this paper in that none of the work has considered VPI.

Recently, we studied the performance of socket-based protocols when using quality-of-service (QoS) provisioning [6]. We found that QoS can have a positive effect on the performance of systems, with greater impact seen in the QoS provisioning of SDP over that of IPoIB.

The performance of the newest PCIe Gen 2 and 40 Gb/s ConnectX adapters was explored by Koop et al. [11]. Narravula et al. investigated the use of caching schemes for InfiniBand-based data center architectures [14]. The performance of InfiniBand networks over long distances was investigated by Rao et al. [16], with a comparison to

10GigE. The authors found that InfiniBand was superior for a single stream of data and that 10GigE was better for multiple streams.

4. Experimental Platform

We conducted experiments on four Dell PowerEdge R805 SMP servers. The PowerEdge R805 has two quad-core 2.0GHz AMD Opteron processors with 12 KB shared execution trace cache, and 16 KB L1 shared data cache on each core. A 512 KB L2 cache for each core and a shared 2 MB L3 cache are available on each chip. There are 8 GB of DDR-2 SDRAM on an 1800 MHz Memory Controller. Each SMP server is equipped with a ConnectX 4X DDR InfiniBand HCA from Mellanox Technologies on a PCIe x8 bus, connected through a 4X InfiniBand DDR Flextronics switch. The ConnectX Ethernet port was tuned according to the guidelines given by Mellanox; IPoIB testing used the same settings where applicable.

The operating system used was a Fedora Core 5 Linux kernel 2.6.20 implementation. The Open Fabrics distribution OFED-1.4 [15] was used as the software stack for InfiniBand. All software was compiled specifically for the machine by using gcc 4.1.1.

5. Performance Results and Analysis

The results shown in Sections 5.1 and 5.2 are from unidirectional tests, while the results in Section 5.3 are based on a real data center and are from bidirectional tests. For the socket-based protocol bandwidth testing, Netperf [10] was used. We required a 99.5% confidence factor in the bandwidth results before a test was considered valid; iperf [19] was used to confirm the results. The latency/bandwidth testing for InfiniBand verbs was done by using qperf from the OFED-1.4 software package. Data center testing was done using the TPC-W benchmark. All bandwidth results are shown in millions of bits per seconds (Mb/s). All message sizes are expressed in Kibibits or Megabits, as applicable.

5.1 Baseline Performance

5.1.1. Single-Stream Tests. The baseline unidirectional latencies of each of the protocols under study are presented in Figure 1. The minimum latency of 1.22 μ s that ConnectX can achieve using InfiniBand RDMA Write is also shown in Figure 1. For the 10GigE tests, we experimented with the *Adaptive Interrupt Coalescing (Receiving)* (AIC-Rx) mechanism to see its impact on Ethernet latency. AIC-Rx allows for the alteration of the behavior of the interrupt generation pattern of a system for incoming traffic. It adapts both the number of frames that must be received before triggering an interrupt and the amount of time from the first packet that is received after triggering an interrupt.

In the case of latency tests, the adaptive algorithm

adjusts the system to create interrupts that fetch the incoming packets according to the traffic pattern, thus increasing the efficiency of interrupts. Since multiple requests are on the transmission line at one time, responding to all of the requests is much faster than if each packet were handled independently or a static coalescing scheme were used. This approach exploits the predictability of the request/response microbenchmark and gives excellent latencies that would be unachievable in most real-world applications. IPoIB does not support adaptive interrupt coalescing, and so IPoIB shows a higher latency for small messages than does AIC-Rx 10GigE. When AIC-Rx is disabled, however, IPoIB shows lower latency.

SDP is superior to 10GigE, with a 65.4% average decrease in latency for messages up to 1 KiB. The reason is that it allows the system to bypass the TCP/IP stack and other software layers and translate socket-based packets directly into the verbs layer RDMA operations [9], while maintaining TCP stream socket semantics.

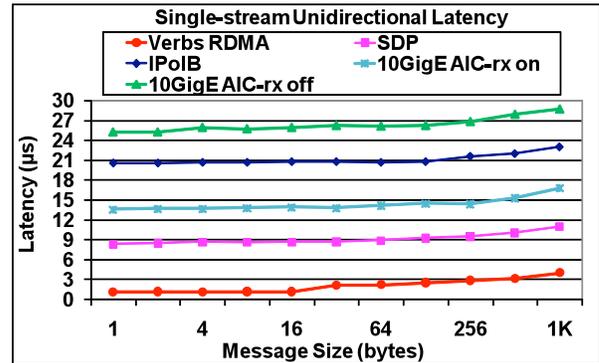


Figure 1. Single-stream latency

Figure 2 illustrates the baseline single-stream bandwidth for 10GigE, IPoIB, SDP, and IB verbs. The IB verbs bandwidth represents the maximum achievable bandwidth for our system of 12650 Mb/s for large messages. We note that the PCIe x8 1.1 practical bandwidth (16 Gb/s theoretical) is a limiting factor on our system.

SDP is the closest in performance to the IB verbs bandwidth, achieving a maximum of ~11880 Mbps. The performance of SDP takes two separate drops, at 4 KiB and at 64 KiB message sizes. The 4 KiB drop point is due to the packet segmentation of the IB fabric of 2 KiB packets; therefore a performance penalty is observed when the additional load required to segment the data stream into smaller message sizes is incurred. The other performance drop occurs at 64 KiB, which is the default threshold value for a switch between buffered-copy and zero-copy transmission methods. This drop can be remedied by adjusting the threshold upwards.

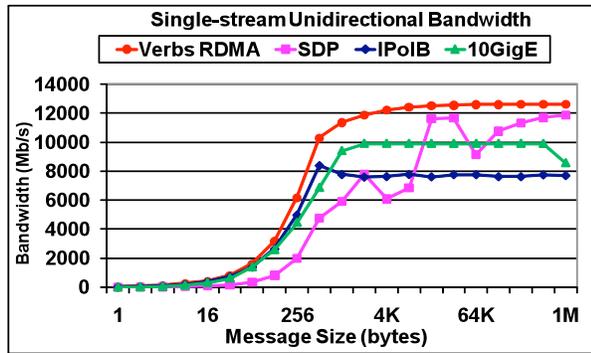


Figure 2. Single-stream bandwidth

Since 10GigE bandwidth with AIC-Rx on or off is similar, we will report the results for the default AIC-Rx from now on as 10GigE in the figures. 10GigE bandwidth is excellent, showing up to 9900 Mb/s for larger message sizes, equivalent to 99% of the maximum theoretical achievable bandwidth. The bandwidth climbs quickly as message size increases, achieving maximum bandwidth at a 2 KiB message size. We observe a dip in performance for 10GigE for message sizes of 1 MiB or larger. This can be resolved by using more than a single connection, as shown in the multistream tests. We note that the results shown in this paper for 10GigE are for jumbo frames; using normal frames, we find the maximal performance to be 28.3% lower for a single stream.

The IPoIB bandwidth is observed to be lower than that for any of the other protocols, with a maximum bandwidth of ~8415 Mb/s at a 512-byte message size. The bandwidth for larger message sizes averages ~7715 Mb/s, and we observe that IPoIB requires more than one thread to achieve maximal throughput. In addition, IPoIB is limited in its single-stream performance by the 2KiB MTU of IB, which outperforms 10GigE using normal frames, with a similar (1500-byte) MTU.

5.1.2. Multistream Tests. The results in Figure 3 show the multistream bandwidths of the 10GigE mode of the ConnectX adapters. The maximum bandwidth is similar to the single-threaded case in Figure 2 at 9900 Mb/s but does not see a drop in performance for 1 MiB messages. We see the expected bandwidth improvement for smaller messages sizes as the number of connections is increased, and no saturation effect occurs at the maximum load (8 streams).

The bandwidth of multistream IPoIB is presented for 2 to 8 simultaneous streams in Figure 4. We can see a great improvement in the multistream IPoIB bandwidth over the single-stream case in Figure 2. The maximum bandwidth peaks at approximately 10800 Mb/s, or 85.4% of maximum verbs bandwidth. This is also significantly better than the single-stream bandwidth, showing a 30.1% improvement.

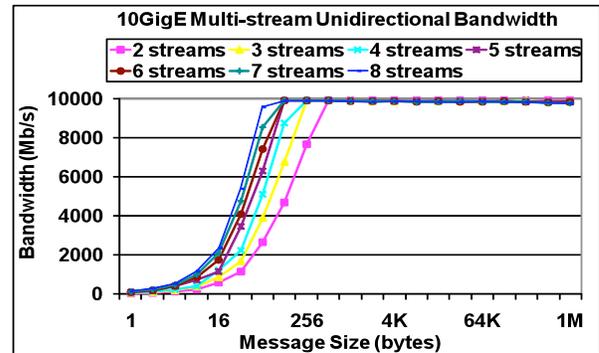


Figure 3. 10GigE multistream bandwidth

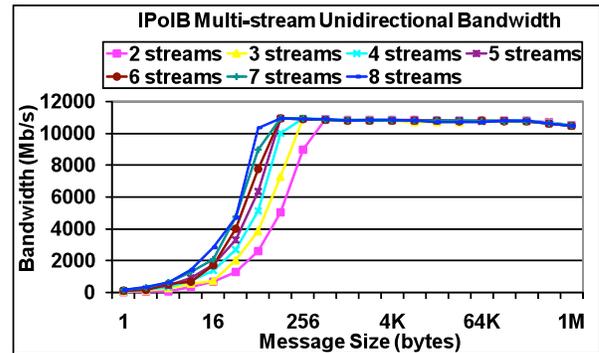


Figure 4. IPoIB multistream bandwidth

The multistream bandwidth for SDP is shown in Figure 5. The maximum bandwidth for SDP occurs with message sizes greater than 128 KiB, showing a maximum bandwidth of approximately 12500 Mb/s. This is only 1.2% lower than that of native IB verbs, showing excellent overall performance.

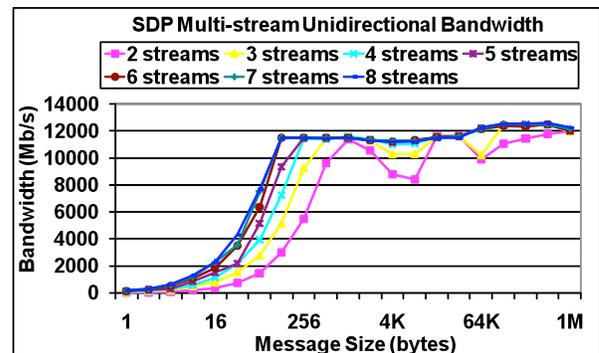


Figure 5. SDP multistream bandwidth

We have found that SDP provides the lowest latencies and the greatest bandwidth. For multistream bandwidth, both IPoIB and SDP have higher available bandwidths than 10GigE, resulting in maximum gains of ~9.6% and 26.2%, respectively. Given that the baseline results are within the expected parameters for the hardware, we are now free to examine other aspects of the system performance and evaluate how such systems might perform in the real world.

5.2 Combined SDP/IPoIB/10GigE VPI Traffic

In this section we explore the effect that IPoIB and SDP traffic over the configured native InfiniBand port has on the performance of the Ethernet traffic operating simultaneously on the other port, and vice versa. Such tests are important in determining the behavior of VPI systems in a hybrid network environment and, to the best of our knowledge, have not been previously carried out. For these tests, even numbers of streams were run in parallel, resulting in four separate pairings, of 1 stream of each up to 4 streams of 10GigE and 4 streams of IPoIB or SDP. 10GigE has been tested with two ports; aggregate bandwidth improvements are insignificant compared to the one-port bandwidth.

The results in Figure 6 show that using combined IPoIB and Ethernet traffic over a single HCA, with different ports for each traffic type, increases the aggregate bandwidth up to ~11300 Mb/s for Ethernet/IPoIB simultaneous traffic, or 10.7% below the verbs maximum and 12.3% greater than 10GigE. There is also an uneven sharing of the available bandwidth between IPoIB and 10GigE, which becomes larger as more streams are utilized. For a single stream, the sharing is even at the highest aggregate bandwidth point. Therefore, for 10GigE there is an uneven sharing of the bus, representing the difference in behavior and efficiency between IPoIB and 10GigE.

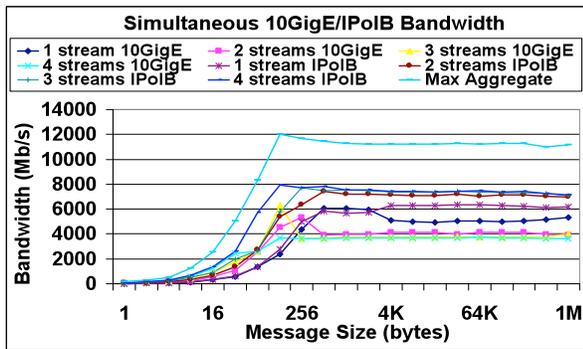


Figure 6. Simultaneous 10GigE/IPoIB bandwidth

The results of simultaneous Ethernet and SDP traffic are illustrated in Figure 7. The figure shows that no significant blocking occurs and both traffic streams can exist harmoniously with an aggregate utilization within 1.2% of verbs bandwidth and 26.7% higher than with 10GigE alone. The effect of bandwidth sharing can be a splitting of bandwidth within 1.2% of perfect division for larger messages with a smaller number of streams. As the number of streams increases, the sharing declines, up to a 2 to 1 ratio of SDP over Ethernet, which will be required in future applications with PCIe Gen2. As was the case for IPoIB, the fairness of sharing the available bandwidth

declines as the number of streams increases.

The results in this section have shown that SDP and Ethernet can harmoniously share a large proportion of the available bandwidth without causing any blocking to their simultaneous traffic partner protocol. IPoIB and Ethernet can also share the bus effectively, but to a lesser degree than that of SDP/Ethernet.

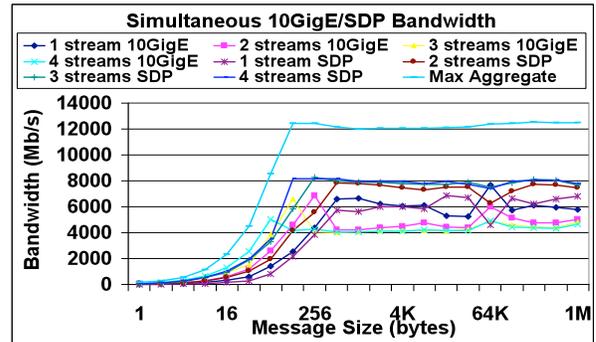


Figure 7. Simultaneous 10GigE/SDP bandwidth

5.3 Data Center Evaluation with VPI

In this section, we explore the use of a three-tier data center architecture consisting of an Apache 2 web server responsible for all static http and image serving, a separate JBoss 5 application server responsible for all server-side Java processing, and a separate database system running MySQL. A high-level overview of the data center implementation used for testing can be seen in Figure 8. This overview also shows the interconnections that were available between each of the given tiers. Note that only the ConnectX card in the web server is configured to operate in the VPI mode.

To assess the performance of VPI in the real world, we used the TPC-W [21] benchmark, which uses a real web-based bookstore to test the latency seen by multiple clients as a total request/response time seen at the client side. It also monitors the average number of web interactions per second that the system is able to sustain, which is a measure of throughput. Strict timing requirements ensure that the system is providing reasonable service to all of its clients. The results shown in this paper are free of any errors during the entire measurement period.

The TPC-W benchmark uses specific load mixes to approximate real data center usage patterns. It utilizes static and dynamic website content, as well as databases for data storage on the server side. The benchmark can be scaled in size, and for our systems it has been implemented using 100,000 items, with the remainder of the data scaled (customers, etc.) as dictated in the specifications. The website is interacted with by a remote browser emulator, which generates multiple clients that interact with the benchmark website. These clients have specified interaction wait times and follow patterns specific to a given behavior set to replicate real

conditions.

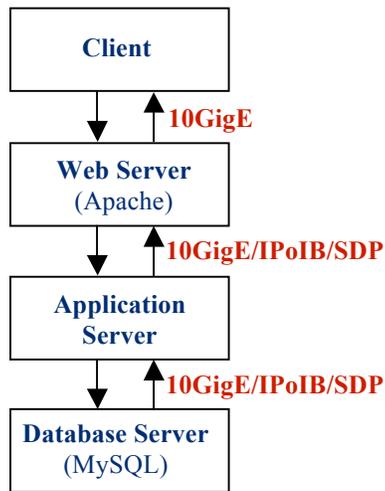


Figure 8. Test data center architecture

The TPC-W implementation used was from New York University [20] and used Enterprise Java beans. It was modified to work with our systems and with updated JBoss and MySQL interfaces. The client load was created by using a remote browser emulator implementation from the University of Wisconsin [7], which was extensively modified to work with our EJB website implementation.

All the results shown in this section use Ethernet connections using jumbo frames to connect the client to the web server. The connections between the web server, the application server, and the database server then used the specified interconnect.

To evaluate the performance results from the varying data center architectures, we must first compare the overall throughput of the three types, shown in Figure 9. The data center setups were run with the highest workload possible that did not generate any errors over the execution period. For the testing period, delineated by the vertical blue lines, the IPoIB data center architecture had the highest throughput, averaging 87.15 interactions per second. SDP has a throughput of 85.08 interactions per second for the given workload mixture. The all-Ethernet architecture has a slightly lower throughput of 83.06 interactions per second. The increased throughput performance of the IPoIB and SDP data center architectures is expected because they provide higher available bandwidths than does Ethernet technology. This performance increase is achievable only because of improvements in data center intracommunication.

Pavan – I don't understand this sentence. The latency results in Figure 9 show the percentage of times a certain request takes longer than the specified response time. Therefore, smaller numbers are better for any given time. Comparing Figure 9(a) with Figure 9(b) for latency results, we can see that the latency difference between an

all-Ethernet and IPoIB/Ethernet VPI data center is small, with IPoIB/Ethernet having an advantage. IPoIB/Ethernet shows benefits for larger operations, such as the new products display, that requires significant interaction between the application server and the database server. There are also improvements in the buy request action, the shopping cart request, and the display of the home page. Therefore, there is a net benefit to using an Ethernet/IPoIB-based VPI data center because it outperforms the all-Ethernet case. This is best seen in the most back-end-intensive requests.

Comparing Figure 9(a) with Figure 9(c) for latency results, we find that the Ethernet/SDP VPI data center in fact provides the best latencies for order display and admin functions, illustrating that the most intensive operations benefit from using an Ethernet/SDP VPI data center configuration. However, we also see higher latencies for functions that are not very back-end server processing heavy. The given load created for the TPC-W browsing mix will not necessarily guarantee that all of the nodes in the data center are continually loaded to levels greater than the interconnect capacity. When the network is not fully loaded, SDP cannot take advantage of pipelining to reduce overall latency. Since the data center utilizes many connections between the given tiers (as many as 250 between the application server and web server), even though the outgoing traffic may be sufficient to load a single outgoing connection, the load is distributed among many connections, and pipelining is not as effective as it would be with fewer connections. Because SDP operates over RDMA, the overhead of the required control messages can cause additional latency that would not otherwise be seen over a heavily loaded connection, where such control messages can be hidden in the transmission latencies of incoming/outgoing traffic by pipelining the RDMA control messages.

An additional factor that might be affecting the system performance is the poor performance of Java's networking protocols over InfiniBand. This reduces the potential benefit from the application server layer because it relies on Java for dynamic processing. The impact of Java on InfiniBand performance was first observed in [22]. Future native support for SDP in the upcoming Sun JRE 7 should help enhance future Java/InfiniBand performance.

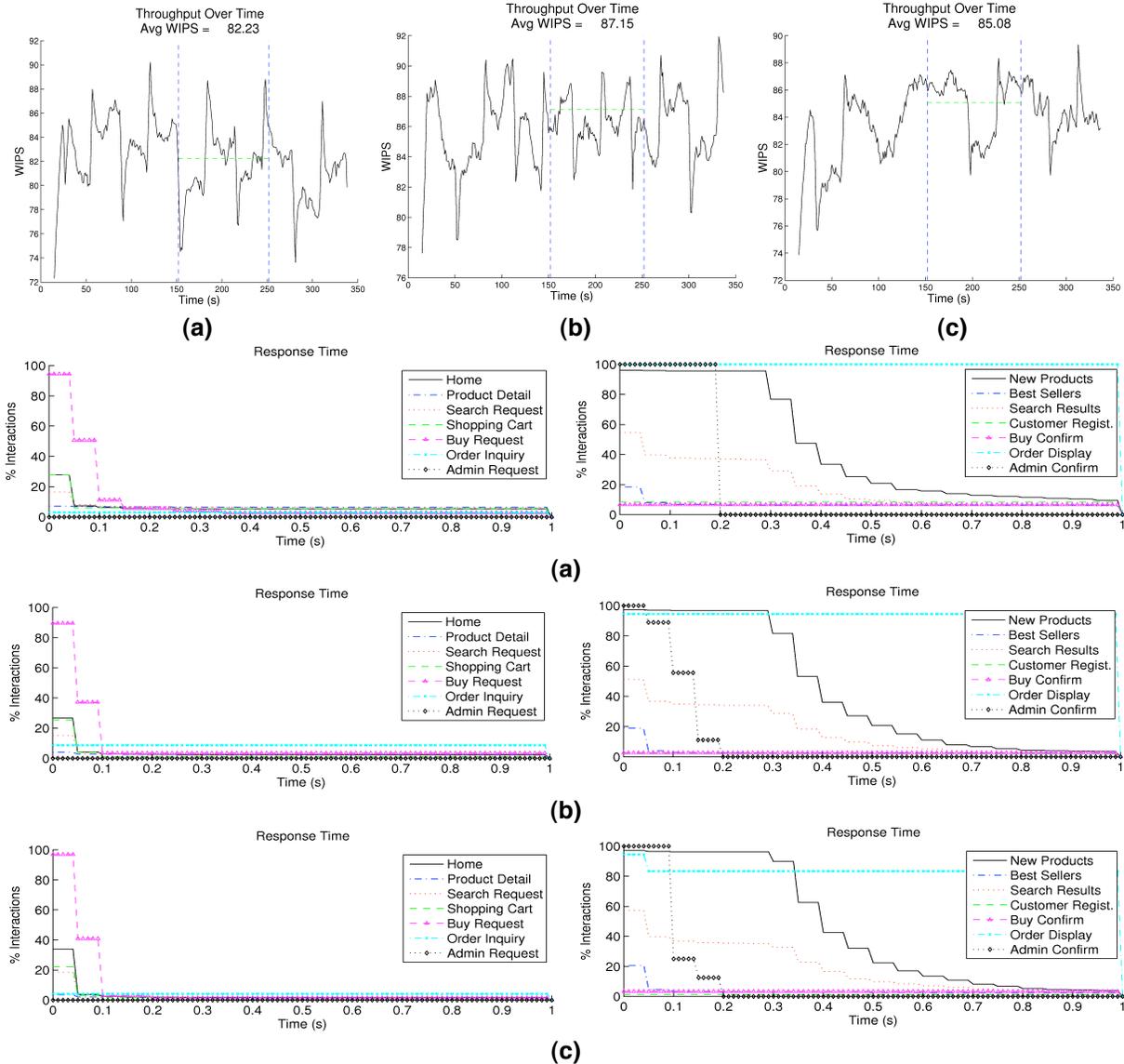


Figure 9. TPC-W data center throughput and latency: (a) all 10GigE, (b) 10GigE/IPoIB, and (c) 10GigE/SDP

Pavan – why is (c) the only one with a y-axis starting at 72? Examining the results of the data center tests and combining the observations with the 10GigE/IPoIB and 10GigE/SDP simultaneous bandwidth results in Section 5.2, we conclude that hybrid data center architectures and VPI can be of benefit in a dynamic web serving content system, where intradata center communication latency and bandwidth are important. In addition, the use of SDP can significantly reduce computational overhead due to network processing in systems [2], which is a factor in increasing the overall system throughput.

6. Conclusions and Future Work

In this paper, we have assessed the basic performance metrics and the effectiveness of VPI in data centers. We have observed a 5% increase in achievable throughput

between a 10GigE data center and an IB/10GigE hybrid data center solution, while providing shorter overall delays, where the avoidance of unacceptable delays is crucial to a successful commercial venture. We have also observed good bandwidth sharing when running multiple protocols simultaneously, providing performance 1.2% below that of IB verbs while increasing aggregate bandwidth by 26.7% overall for 10GigE alone; latency can be reduced by 54.4% on average. IPoIB/10GigE was found to be the best configuration for a real-world data center test and performed well in bandwidth and latency tests, making it the preferred configuration for current firmware and driver releases. VPI represents a step forward in the performance of data centers, and much of this potential gain should be realizable through future optimization of data center software and system

configurations.

In addition to increased performance, VPI also allows access to a variety of other very useful features in InfiniBand networks. The advanced quality of service mechanisms available for IB networks can be used for greater performance and traffic control. Our observations were limited by the available bus bandwidth and should scale when using higher-speed bus technology such as PCI Gen 2 and new 40 Gb/s InfiniBand adapters. Hybrid data centers could be of great use for streaming based applications where very large bandwidth requirements exist.

Our future work in the area of VPI will concentrate on improving the performance of VPI-based data centers, using the available InfiniBand network features to enhance performance. In addition, we will investigate the improvement of SDP in a data center environment, particularly as it relates to latencies, and we will explore the behavior of different applications.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada Grant #RGPIN/238964-2005; the Canada Foundation for Innovation and Ontario Innovation Trust Grant #7154; the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357; and the National Science Foundation Grant #0702182. We thank Mellanox Technologies for the resources.

References

- [1] P. Balaji, S. Bhagvat, H. Jin and D.K. Panda, "Asynchronous zero-copy communication for synchronous sockets in the Sockets Direct Protocol (SDP) over InfiniBand," in *6th Workshop on Communication Architecture for Clusters (CAC)*, 2006.
- [2] 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 *IEEE Int. Symp. on Performance Analysis of Systems and Software (ISPASS)*, 2004, pp. 10-12.
- [3] J. Beecroft, D. Addison, D. Hewson, M. McLaren, D. Roweth, F. Petrini and J. Nieplocha, "QsNet^{II}: Defining high-performance network design," *IEEE Micro*, 25(4), pp. 34-47, 2005.
- [4] ConnectX VPI product brief, Mellanox Technologies, [http://www.mellanox.com/related-docs/prod_adapter_cards/PB_ConnectX_VPI.pdf].
- [5] D. Goldenberg, M. Kagan, R. Ravid, and M. Tsirkin, "Transparently achieving superior socket performance using zero copy Socket Direct Protocol over 20Gb/s InfiniBand links," in *2005 IEEE Int. Conf. on Cluster Computing (Cluster)*, pp. 1-10.
- [6] R.E. Grant, M.J. Rashti and A. Afsahi, "An analysis of QoS provisioning for Sockets Direct Protocol vs. IPoIB over modern InfiniBand networks," in *Int. Workshop on Parallel*

Programming Models and Systems Software for High-End Computing (P2S2), 2008, pp. 79-86.

- [7] T. Horvath, TPC-W Java client implementation. [<http://www.ece.wisc.edu/~pharm/tpcw.shtml>], 2008.
- [8] W. Huang, J. Han, J. He, L. Zhang, and Y. Lin, "Enabling RDMA capability of InfiniBand network for Java applications," in *Int. Conf. on Networking, Architecture, and Storage (NAS)*, 2008, pp. 187-188.
- [9] InfiniBand Trade Association. InfiniBand Architecture Specification, Volume 1, October 2004.
- [10] R. Jones, Netperf, 2008.
- [11] M.J. Koop, W. Huang, K. Gopalakrishnan, and D.K. Panda, "Performance analysis and evaluation of PCIe 2.0 and quad-data rate InfiniBand," in *16th IEEE Symp. on High Performance Interconnects (HotI)*, 2008, pp. 85-92.
- [12] Mellanox Technologies, [<http://www.mellanox.com>].
- [13] Myricom, [<http://www.myricom.com>].
- [14] S. Narravula, H.W. Jin, K. Vaidyanathan, and D.K. Panda, "Designing efficient cooperative caching schemes for multi-tier data-centers over RDMA-enabled networks," in *6th IEEE/ACM Int. Symp. on Cluster Computing and the Grid (CCGRID)*, 2006, pp. 401-408.
- [15] OpenFabrics Alliance, [<http://www.openfabrics.org/>].
- [16] N.S.V. Rao, W. Yu, W.R. Wing, S.W. Poole, and J.S. Vetter, "Wide-area performance profiling of 10GbE and InfiniBand technologies," in *2008 ACM/IEEE Conf. on Supercomputing (SC)*, 2008.
- [17] RDMA Consortium. iWARP protocol specification. [<http://www.rdmaconsortium.org/>].
- [18] H.V. Shah, C. Pu, and R.S. Madukkarumukumana, "High performance sockets and RPC over Virtual Interface (VI) architecture," in *3rd Workshop on Network-Based Parallel Computing: Communication, Architecture, and Applications (CANPC)*, 1999, pp. 91-107.
- [19] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf: The TCP/UDP bandwidth measurement tool," [<http://dast.Nlanr.net/Projects/Iperf/>], 2004.
- [20] A. Totok, TPC-W-NYU. [<http://cs.nyu.edu/~totok/professional/software/tpcw/tpcw.html>], 2005.
- [21] TPC-W Benchmark, [<http://www.tpc.org/tpcw/>].
- [22] H. Zhang, W. Huang, J. Han, J. He, and L. Zhang, "A performance study of Java communication stacks over InfiniBand and giga-bit Ethernet," in *IFIP Int. Conf. on Network and Parallel Computing Workshops (NPC)*, 2007, pp. 602-607.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.