# Micro-Benchmark Level Performance Comparison of High-Speed Cluster Interconnects [*]

Jiuxing Liu    Balasubramanian Chandrasekaran    Weikuan Yu    Jiesheng Wu
Darius Buntinas    Sushmitha Kini    Peter Wyckoff[†]    Dhabaleswar K. Panda

Computer and Information Science
The Ohio State University
Columbus, OH 43210

[†]Ohio Supercomputer Center
1224 Kinnear Road
Columbus, OH 43212

## Abstract

*In this paper, we present a comprehensive performance evaluation of three high speed cluster interconnects: InfiniBand, Myrinet and Quadrics. We propose a set of micro-benchmarks to characterize different performance aspects of these interconnects. Our micro-benchmark suite includes not only traditional tests and performance parameters, but also those specifically tailored to the interconnects' advanced features such as user-level access for performing communication and remote direct memory access. In order to explore the full communication capability of the interconnects, we have implemented the micro-benchmark suite at the low level messaging layer provided by each interconnect. Our performance results show that all three interconnects achieve low latency, high bandwidth and low host overhead. However, they show quite different performance behaviors when handling completion notification, unbalanced communication patterns and different communication buffer reuse patterns.*

## 1 Introduction

Today's distributed and high performance applications require high computational power as well as high communication performance. In the past few years, the computational power of commodity PCs has been doubling about every eighteen months. At the same time, network interconnects that provide very low latency and very high bandwidth are also emerging. This trend makes it very promising to build high performance computing environments by clustering, which combines the computational power of commodity PCs and the the communication performance of high speed network interconnects.

Currently, there are several network interconnects that provide low latency (less than 10us) and high bandwidth (in the order of Gbps). Two of the leading products are Myrinet[10] and Quadrics[11]. More recently, InfiniBand[7] has entered the high performance computing market. All three interconnects share many similarities: They provide user-level access to the network interface card for performing communication and support access to remote node's memory space. However, they also differ in a lot of ways. Therefore, an interesting question arises: *How can we conduct a meaningful performance comparison among all the three interconnects?*

Traditionally, simple micro-benchmarks such as latency and bandwidth tests have been used to characterize the communication performance of network interconnects. Later, more sophisticated models such as LogP[5] were proposed. However, these tests are designed for general parallel computing systems and they do not address many features that are present in the interconnects studied in this paper.

Another way to evaluate different network interconnects is to use real world applications. However, real applications usually run on top of a middleware layer such as the Message Passing Interface (MPI). Therefore, the performance we see reflects not only the capability of the network interconnects, but also the quality of the MPI implementation and the design choices made by different MPI implementers. Thus, to provide more insights into the communication capability offered by each interconnect, it is desirable to conduct tests at a lower level.

In this paper, we have proposed an approach to evaluate and compare the performance of three high speed interconnects: InfiniBand, Myrinet and Quadrics. We have designed a set of micro-benchmarks to characterize different aspects of the interconnects. Our micro-benchmarks include not only traditional performance measurements, but also tests that are more relevant to networks that provide user-level mode access. Our benchmarks also concentrate on the remote memory access capabilities provided by each interconnect. We have conducted tests in an 8-node cluster that has all the three network interconnects. From the experiments we have found that although these interconnects have similar programming interfaces, their performance behavior differs significantly when it comes to handling completion notification, different buffer reuse patterns and unbalanced communication, all of which cannot be evaluated by latency/bandwidth tests.

The rest of this paper is organized as follows: In Section 2 we provide an overview of the studied interconnects and their messaging software. In Section 3 we present our micro-benchmarks and their performance results. We then

[*] This research is supported in part by Sandia National Laboratory's contract #30505, Department of Energy's Grant #DE-FC02-01ER25506, and National Science Foundation's grants #EIA-9986052 and #CCR-0204429.

discuss related work in Section 4. We draw conclusions in Section 5.

# 2 Overview of Interconnects

## 2.1 InfiniBand/VAPI

The InfiniBand Architecture defines a System Area Network (SAN) for interconnecting processing nodes and I/O nodes. In an InfiniBand network, processing nodes and I/O nodes are connected to the fabric by Host Channel Adapters (HCAs) or Target Channel Adapters (TCAs).

Our InfiniBand platform consists of InfiniHost HCAs and an InfiniScale switch from Mellanox[8]. InfiniScale is a full wire-speed switch with eight 10 Gbps ports. The InfiniHost MT23108 HCA connects to the host through PCI-X bus. It allows for a bandwidth of up to 10 Gbps over its ports.

VAPI is the software interface for InfiniHost HCAs. The interface is based on InfiniBand verbs layer. It supports both send/receive operations and remote direct memory access (RDMA) operations. Currently, Reliable Connection (RC) and Unreliable Datagram (UD) services are implemented. In this paper, we concentrate on RC service. In VAPI, user buffers must be registered first before they can be used for communication. The completion of communication requests is reported through completion queues (CQs).

## 2.2 Myrinet/GM

Myrinet was developed by Myricom[9] based on communication and packet-switching technology originally designed for massive parallel processors (MPPs). Myrinet has a user-programmable processor on the network interface card that provides much flexibility in designing communication software.

Our Myrinet network consists of Myrinet-2000 network interface cards connected by a Myrinet-2000 switch. The link bandwidth of the Myrinet network is 2Gbps. The Myrinet-2000 switch is a 8-port crossbar switch. The network interface card has a 66MHz/64bit PCI interface. It has a programmable Lanai 9.3 processor running at 200 MHz with 2MB on-board SRAM. The Lanai processor on the NIC can access host memory via the PCI bus through the DMA controller.

GM is the low-level messaging layer for Myrinet clusters. It provides protected user-level access to the network interface card and ensures reliable and in-order message delivery. GM provides to the upper layer a connectionless communication model. GM supports send/receive operations. It also has directed send operation which can directly write data to a remote node's address space. Similar to VAPI, user communication buffers must be registered in GM.

## 2.3 Quadrics/Elanlib

Quadrics networks consist of Elan3 network interface cards and Elite switches[13]. The Elan network interface cards are connected to hosts via 66MHz/64Bit PCI bus. Elan3 has 64 MB on-board SDRAM and a memory management unit (MMU). An Elite switch uses a full crossbar connection and supports wormhole routing.

Our Quadrics network consists of Elan3 QM-400 network interface cards and an Elite 16 switch. The Quadrics network has a transmission bandwidth of 400Mbytes/s in each link direction.

Elanlib supports protected, user-level access to Elan network interfaces. It provides a global virtual address space by integrating individual node's address space. One node can use DMA to access a remote node's memory space. Elanlib provides a general-purpose synchronization mechanism based on events stored in Elan memory. The completion of remote DMA operations can be reported through events. Unlike VAPI and GM, communication buffers do not need to be registered. Elan network interface cards have an on-board memory management unit. The system software is responsible for synchronizing the MMU table and doing the address translation.

# 3 Micro-Benchmarks and Performance

To provide more insights into communication behavior of the three interconnects, we have designed a set of micro-benchmarks and performance parameters to reveal different aspects of their communication performance. They include traditional benchmarks and performance parameters such as latency, bandwidth and host overhead. In addition, we have designed tests that are more related to the user-level communication model used by these interconnects. For example, we have designed benchmarks to characterize the impact of address translation mechanisms used by the network interface cards as well as the effect of completion notification. In the hot-spot tests, we provide information about how these interconnects can handle unbalanced communication patterns. We also quantify the performance impact of PCI-X 133MHz bus used by our InfiniBand network.

Our experimental testbed consists of 8 SuperMicro SUPER P4DL6 nodes with ServerWorks GC chipsets and dual Intel Xeon 2.40 GHz processors. The machines were connected by InfiniBand, Myrinet and Quadrics interconnects. The InfiniHost HCA adapters work under the PCI-X 64-bit 133MHz interfaces. The Myrinet and Quadrics cards use 64-bit 66MHz PCI slots. We used the Linux Red Hat 7.2 operating system.

## 3.1 Latency and Bandwidth

End-to-end latency has been frequently used to characterize the performance of interconnects. All the interconnects under study support access to remote node's memory space. We thus also measured the latency to finish a remote put operation. InfiniBand/VAPI and Myrinet/GM also support send/receive operations. Figure 1 shows the latency results. For small message, Elanlib achieves the best latency, which is around 2$\mu$s. VAPI RDMA latency is around 6$\mu$s and send/receive latency is around 8$\mu$s. GM has a latency of about 7$\mu$s for small messages. For messages less than 64 bytes, GM send/receive can combine data and descriptor at the sender side. Therefore it offers better performance than directed send.

The bandwidth test is used to determine the maximum sustained data rate that can be achieved at the network level. In this test, a sender keeps sending back-to-back messages to the receiver until it has reached a pre-defined queue size Q. Then it waits for Q/2 messages to finish and sends out another Q/2 messages. In this way, the sender ensures that there are at least Q/2 and at most Q outstanding messages. A similar method has been used in [4]. Figure 2 shows the bandwidth results with very large queue size. Figure 3 shows the bandwidth with different queue size. The peak bandwidth for VAPI, GM, and Quadrics is around 820MB/s, 220MB/s and 320MB/s, respectively. We can see that VAPI is more sensitive to the value of queue size Q and it performs much better for large messages. (Note that unless stated otherwise, the unit MB in this paper is an abbreviation for $2^{20}$ bytes.)

### 3.2 Bi-Directional Latency and Bandwidth

Compared with uni-directional latency and bandwidth tests, bi-directional latency and bandwidth tests put more stress on the PCI bus, the network interface cards, and the switches. Therefore they may be more helpful to us to understand the bottleneck in communication. The tests are carried out in a way similar to the uni-directional tests. The difference is that both sides send data simultaneously. From Figure 4, we can see bi-directional latency performance for all three interconnects is worse than their uni-directional latency. Figure 5 also shows similar results for bandwidth. We see that for VAPI, the PCI-X bus becomes the bottleneck and limits the bandwidth to around 900MB/s. Although Quadrics has better uni-directional bandwidth than GM, its peak bi-directional bandwidth is only around 360MB/s, which is less than GM's 400MB/s.

### 3.3 Host Communication Overhead

We define host communication overhead as the time CPU spends on communication tasks. The more time CPU spends in communication, the less time it can do computation. Therefore this can serve as a measurement of the ability of a messaging layer to overlap communication and computation. We measure the host overhead for both latency and bandwidth tests. In the latency test, we directly measure the CPU overhead for different message sizes. In the bandwidth test, we insert a computation loop in the program. By increasing the time of this computation loop, eventually we see a drop in the bandwidth.

Figure 6 presents the host overhead in the latency test. VAPI has the highest overhead, which is around 2$\mu$s. Elanlib overhead is around 0.7$\mu$s. GM has the least overhead, which is around 0.5$\mu$s. GM reduces the overhead further for messages less than 64 bytes by combining data and send descriptors. Figure 7 shows the impact of computation time on bandwidth. All three interconnects can overlap communication and computation quite well. VAPI bandwidth drops only after over 95% of running time is used for computation. GM and Quadrics bandwidth drops only after more than 99%.

### 3.4 Overhead of Completion Notification

Since all three interconnects support remote direct memory access, one way to detect the arrival of messages at the receiver side is to poll on the memory content in the destination buffer. This approach can be used to minimize the receiver overhead. However, this method is hardware dependent because it relies on the order how the DMA controller writes to host memory.

The network interconnects we have studied support different mechanisms to report the completion of remote memory operations. For example, VAPI uses CQ, while GM and Elanlib rely on event abstractions. Figure 8 shows overhead of using these mechanisms for remote memory access at the receiver side. Quadrics has very efficient notification mechanism, which adds only 0.4$\mu$s overhead for large messages. For messages less than 64 bytes, there is no extra overhead. VAPI has an overhead of around 1.8$\mu$s. GM's directed send does not have a mechanism to notify the receiver of message arrival. Therefore, we simulated the notification by using a separate send operation. This adds around 3-5$\mu$s overhead. Instead of busy polling, the upper layer can also use blocking to wait for completions. From Figure 9 we can observe that VAPI has the highest overhead, which is over 20$\mu$s. The overheads for GM and Quadrics are about 11$\mu$s and 13$\mu$s, respectively.

### 3.5 Impact of Buffer Reuse

In most micro-benchmarks that are designed to test communication performance, only one buffer is used at the sender side and the receiver side, respectively. However, in real applications a large number of different buffers are usually used for communication. The buffer reuse pattern can have a significant impact on the performance of interconnects that support user-level access to network interfaces such as those studied in this paper.

To capture the cost of address translation at the network interface, we have designed two schemes of buffer reuse pattern and we have changed the tests accordingly. In the first scheme, N different buffers of the same size are used in FIFO order for multiple iterations. By increasing the number N, it may happen that eventually the performance drops. Basically, this test measures how large the communication working set can be in order to get best communication performance. Figure 10 shows the bandwidth results with 512KBytes messages. We can see that up to 100 buffers, GM and Quadrics show no performance degradation. However, VAPI performance drops when more than 10 buffers are used.

The second scheme is slightly more complicated. In this scheme, the test consists of N iterations and we define a buffer reuse percentage R. For the N iterations of the test, N*R iterations will use the same buffer, while all other iterations will use completely different buffers. By changing buffer reuse percentage R, we can see how communication performance is affected by buffer reuse patterns. From Figures 11 and 12, we can see that Quadrics is very sensitive to buffer reuse patterns. Its performance drops significantly when the buffer reuse rate decreases. VAPI also shows the similar behavior. GM latency performance drops when the buffer reuse rate decreases, but its bandwidth performance is not sensitive to the buffer reuse rate.

### 3.6 Hot-Spot Tests

Hot-spot tests are designed to measure the ability of network interconnects to handle unbalanced communication patterns. Similar tests have been conducted in [12]. We have used two sets of hot-spot tests. In hot-spot send tests, a master node keeps sending messages to a number of different slave nodes. In hot-spot receive tests, the master node receives messages from all the slave nodes. We vary the number of slave nodes. Figures 13 and 14 show the hot spot performance results. We can see that Quadrics scales very good when the number of slaves increases. On the other handle, VAPI and GM do not scale very well.

### 3.7 Impact of PCI and PCI-X Bus on VAPI

In previous comparisons, our InfiniHost HCAs use PCI-X 133 MHz interface while Myrinet and Quadrics only support PCI 66 MHz interface. To see how much impact the PCI-X interface has on the communication performance of InfiniBand, we have forced the InfiniHost HCAs to use PCI interface running at 66 MHz. Figures 15 and 16 compare VAPI RDMA write latency and bandwidth with PCI and PCI-X configuration. (UD stands for Uni-Directional and BD stands for Bi-Directional in the figures.) With PCI, there is a slightly performance degradation for small messages. For large messages, the performance drops dramatically due to the limited bandwidth of PCI bus.

## 4 Related Work

LogP [5] and its extension LogGP [1] are the methodologies which are often used to extract performance parameters in conventional communication layers. In addition to the performance parameters shown in the LogGP model, our study explores performance impact of other advanced features that are present in the interconnects under study.

Studies on the performance of communication layers on the Myrinet network and the Quadrics network have been carried out in the literature [2, 12, 11]. Our previous work [3] devises a test suite and uses it to compare performance for several VIA[6] implementations. In this paper, we extend the work by adding several important scenarios which have strong application implication and apply them to a wider range of communication layers and networks. Bell *et al* [4] evaluate performance of communication layers for several parallel computing systems and networks. However, their evaluation is based on the LogGP model and they have used different testbeds for different interconnects.

## 5 Conclusions

In this paper, we have used a set of micro-benchmarks to evaluate three high performance cluster interconnects: InfiniBand, Myrinet and Quadrics. We provide a detailed performance evaluation for their communication performance by using a set of micro-benchmarks. We show that in order to get more insights into the performance characteristics of these interconnects, it is important to go beyond simple tests such as latency and bandwidth. Specifically, we need to consider the performance impact of certain features such as remote memory access, completion notification and address translation mechanisms in the network interface.

## References

[1] A. Alexandrov, M. F. Ionescu, K. E. Schauser, and C. Scheiman. LogGP: Incorporating long messages into the LogP model for parallel computation. *Journal of Parallel and Distributed Computing*, 44(1):71–79, 1997.

[2] S. Araki, A. Bilas, C. Dubnicki, J. Edler, K. Konishi, and J. Philbin. User-space communication: A quantitative study. In *SC98: High Performance Networking and Computing*, November 1998.

[3] M. Banikazemi, J. Liu, S. Kutlug, A. Ramakrishna, P. Sadayappan, H. Shah, and D. K. Panda. VIBe: A Micro-benchmark Suite for Evaluating Virtual Interface Architecture (VIA) Implementations. In *IPDPS*, April 2001.

[4] C. Bell, D. Bonachea, Y. Cote, J. Duell, P. Hargrove, P. Husbands, C. Iancu, M. Welcome, and K. Yelick. An evaluation of current high-performance networks. In *International Parallel and Distributed Processing Symposium (IPDPS'03)*, April 2003.

[5] D. E. Culler, R. M. Karp, D. A. Patterson, A. Sahay, K. E. Schauser, E. Santos, R. Subramonian, and T. von Eicken. Logp: Towards a realistic model of parallel computation. In *Principles Practice of Parallel Programming*, pages 1–12, 1993.

[6] D. Dunning, G. Regnier, G. McAlpine, D. Cameron, B. Shubert, F. Berry, A. Merritt, E. Gronke, and C. Dodd. The Virtual Interface Architecture. *IEEE Micro*, pages 66–76, March/April 1998.

[7] InfiniBand Trade Association. InfiniBand Architecture Specification, Release 1.0, October 24 2000.

[8] Mellanox Technologies. Mellanox InfiniBand InfiniHost Adapters, July 2002.

[9] Myricom. Myricom, Inc. http://www.myri.com.

[10] N.J. Boden, D. Cohen, R.E. Felderman, A.E. Kulawik, C.L. Seitz, J.N. Seizovic, and W. Su. Myrinet: A Gigabit-per-second Local Area Network. *IEEE Micro*, 15(1):29–36, February 1995.

[11] F. Petrini, W. Feng, A. Hoisie, S. Coll, and E. Frachtenberg. The Quadrics Network: High-Performance Clustering Technology. *IEEE Micro*, 22(1):46–57, 2002.

[12] F. Petrini, A. Hoisie, W. chun Feng, and R. Graham. Performance Evaluation of the Quadrics Interconnection Network. In *Workshop on Communication Architecture for Clusters 2001 (CAC '01)*, April 2001.

[13] Quadrics. Quadrics, Ltd. http://www.quadrics.com.

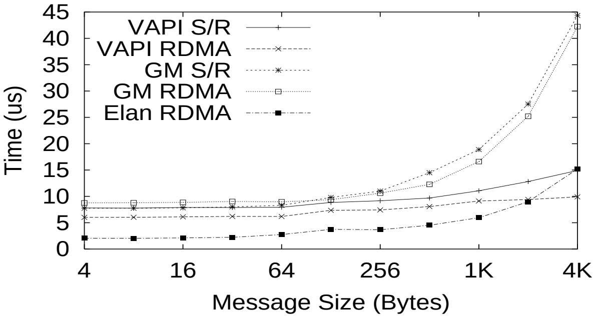

**Figure 1. Latency**

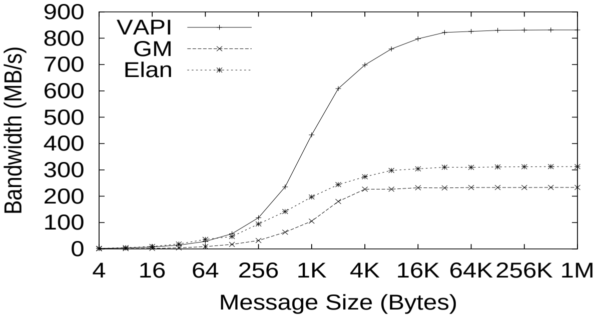

**Figure 2. Bandwidth**

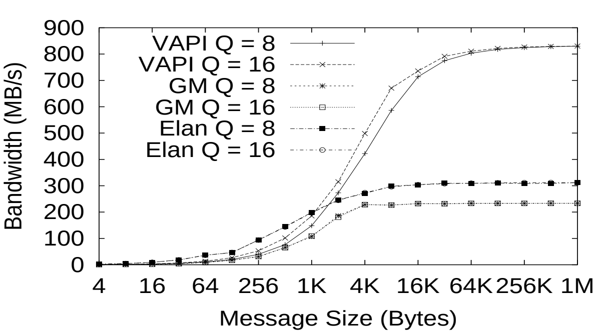

**Figure 3. Bandwidth with Queue Size**

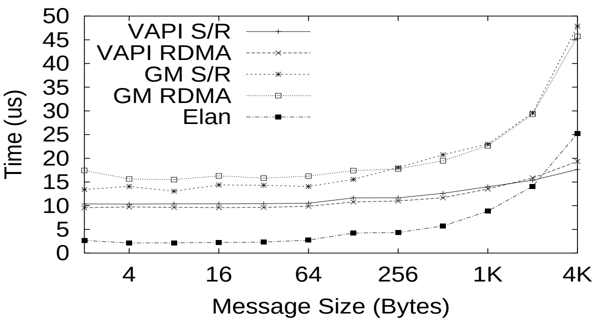

**Figure 4. Bi-Directional Latency**

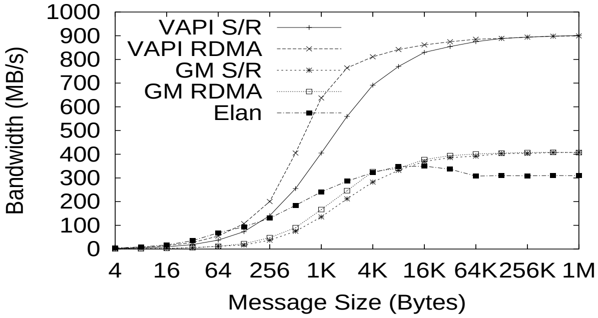

**Figure 5. Bi-Directional Bandwidth**

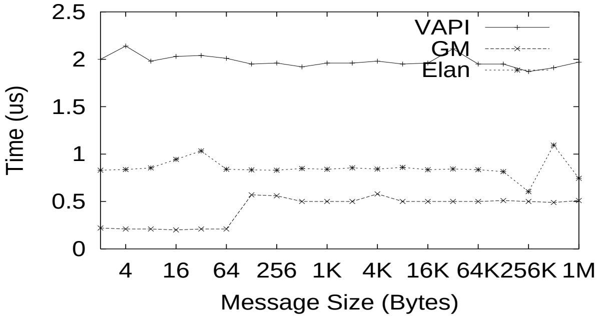

**Figure 6. Host Overhead in Latency Test**

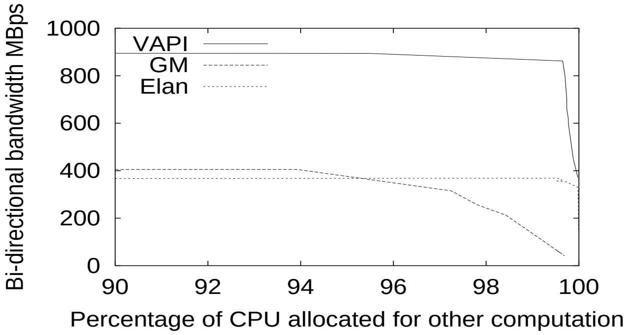

**Figure 7. CPU Utilization in Bandwidth Test**

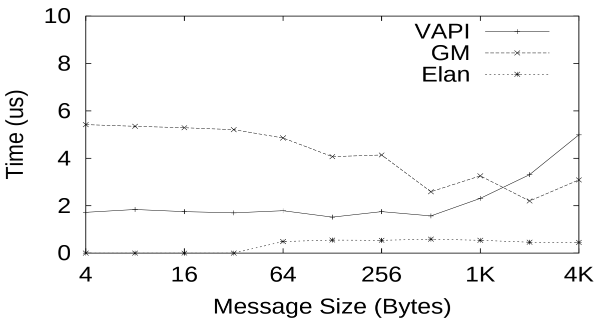

**Figure 8. Overhead due to Completion**

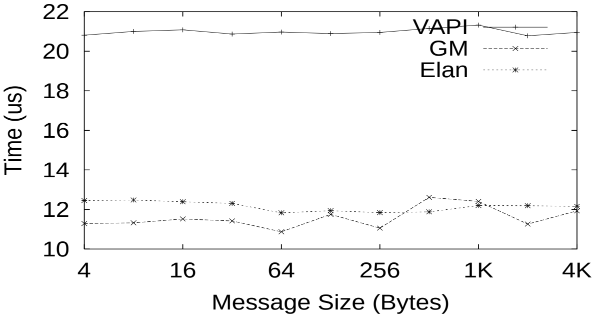

**Figure 9. Overhead due to Blocking**

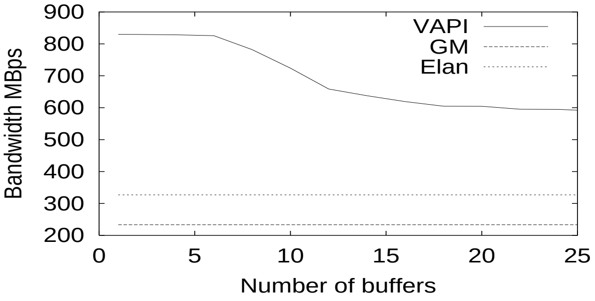

**Figure 10. Bandwidth (size=512K) Buffer Scheme 1**

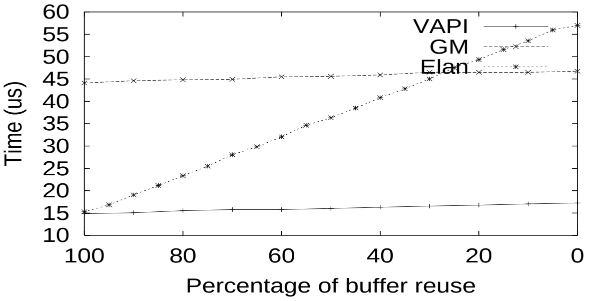

**Figure 11. Latency (size=4K) Buffer Scheme 2**

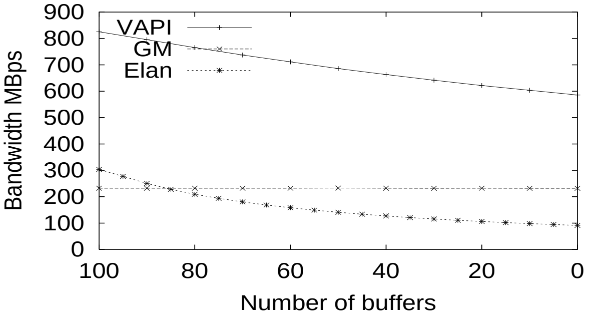

**Figure 12. Bandwidth (size=512K) Buffer Scheme 2**

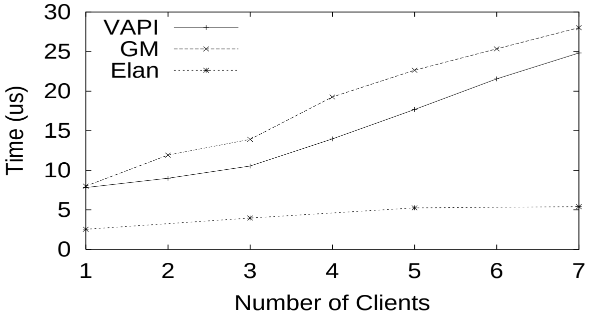

**Figure 13. Hot Spot Test Send**

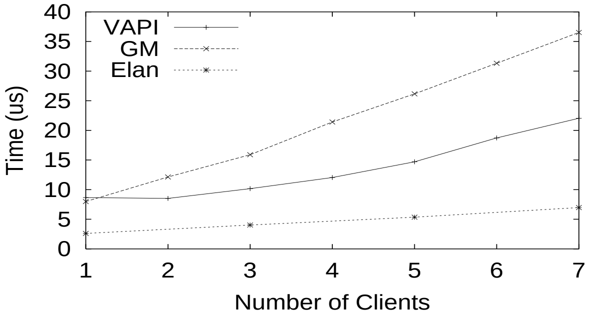

**Figure 14. Hot Spot Test Receive**

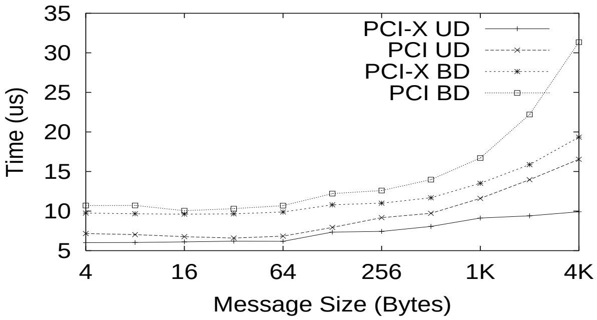

**Figure 15. Impact of PCI on latency**

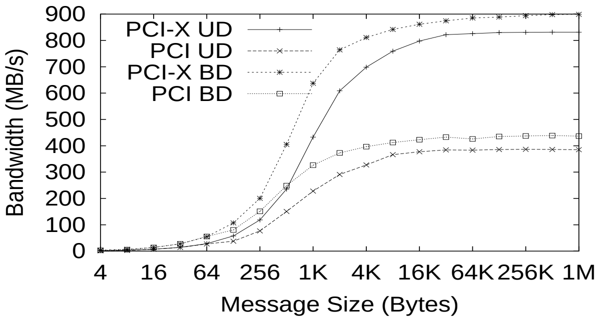

**Figure 16. Impact of PCI on bandwidth**