User Level Communication on Alpha Linux Systems

Jie Chen and William Watson III
High Performance Computing Group
Thomas Jefferson National Accelerator Facility
Newport News, Virginia 23606, U.S.A

Abstract

Recent advances in commodity network interface technology enable scientists and engineers to build clusters of workstations or PCs to execute parallel applications. However, raw-hardware network performance is rarely delivered to applications because of the overheads of communication software and operating systems. To reduce these overheads, a technique called ***user-level*** *communication can be used to allow applications to access the network interface directly without intervention from the operating system. In this paper we examine two user-level communication systems,GM and BIP, on* ***alpha*** *based systems connected by a Myrinet network. In addition quantitative studies on how DMA initiation costs, flow control and reliable communication will effect the performance of communication software are presented.*

1 Introduction

In recent years, we have seen arrival of scalable, low latency interconnection networks such Myrinet[1], the Memory Channel[2] and Giganet[3] for workstation or PC clusters. These "hot interconnects" allow construction of cost effective parallel computing clusters from commodity parts. The cluster approach to high performance computing tracks the technology curve closely, enables executions of some scientific and engineering computing which used to run exclusively on dedicated MPPs and SMPs. However, to achieve comparable results to that of MPPs and SMPs, the cluster communication systems must deliver application-to-application performance at least comparable to that of MPPs.

The performance of these high performance or "hot interconnects" is determined by their network interfaces (NI) and corresponding software components rather than the switches and links. In particular these new interconnects have revealed many problems with the traditional implementation approach toward network communication software: multiple data transfers through the operating system (OS) to applications. For example, the following figure illustrates a typical TCP performance between two Compaq XP1000 (Alpha 21264 CPU) connected by Gigabit Ethernet cards (GA620 from Netgear). The figure clearly demonstrates that the traditional implementation of TCP can only achieve roughly 60 percent of hardware raw performance due to multiple data copying, multiple invocations of system calls and buffer management, even though the performance test is conducted with several new features such as jumbo frame and hardware IP checksum enabled. In addition, OS involvement in data transfer introduces unacceptable high latency for small messages.

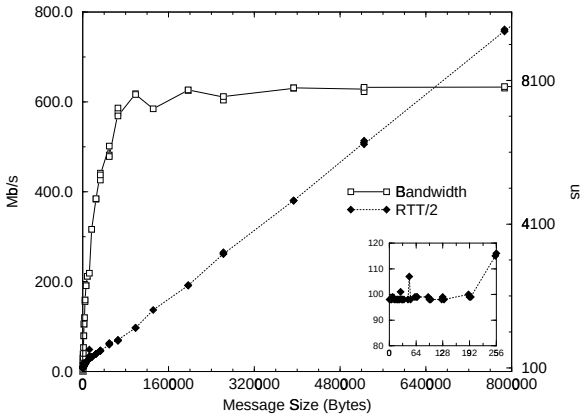

Figure 1. TCP over Gigabit Ethernet.

The **user-level**[4] communication technique addresses problems caused by traditional implementations of communication software. The main causes of degrading network performance through traditional approach are system calls, data buffer management and data copying. To avoid some or all these overheads, several lean user-level communication systems have been implemented by universities and vendors. These systems allow applications to access network

interface directly and remove operating systems from critical paths of send/receive operations. These systems can be used to implement traditional, compatibility libraries such as BSD sockets and MPI, and to write applications directly. Figure 2 is a "copy diagram" that illustrates the difference between traditional and user-level communication software in terms of steps involved in sending/receiving messages from user process to the network.

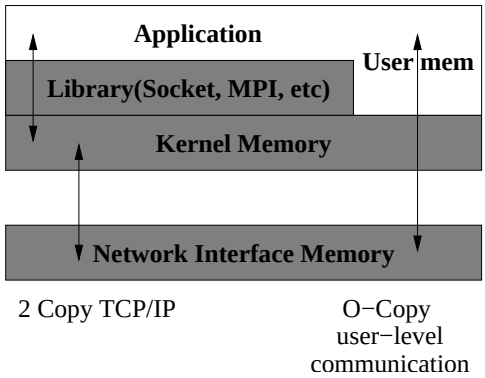

Figure 2. Copy diagram of communication software.

In this paper we investigate two commonly used communication systems, GM 1.1.3 and BIP-0.99 on a cluster of Alpha 21264 based PCs running Linux operating system (2.2.5 kernel) and interconnected by a Myrinet network. The LogP[5] method is used in the investigation to provide insight into how performance is affected by the programming model and implementation choices of each system. In addition an enhancement to reduce latency for small messages together with an option to allow unreliable delivery of small messages is added to GM. The performance data of modified GM (called GM+) will also be presented.

2 Test Configuration

All performance data are collected on a pair of Compaq XP1000 workstations connected directly by a Myrinet network. Each computer has an Alpha 21264 CPU running at 500MHz, 256 MB memory, 64KB L1 cache, 4MB L2 cache and two PCI buses that are capable of doing 64bit or 32bit at 33 MHz. Each computer is running Alpha linux with kernel version 2.2.5.

Myrinet is a Gigabit-per-second (Gb/s) LAN based on the technology used for packet communication in MPPs. The Myrinet technology consists of host interfaces, switches, and 1.28 Gb/s full-duplex point-to-point links. The Myrinet links provide in-order network delivery with low bit error rates (below 10^{-15} on cables up to 25m long).The payload of a Myrinet packet is of arbitrary length. Upon sending, an 8-bit cyclic-redundancy-check (CRC) character is computed on the entire packet and appended to the packet. On packet arrival, the CRC bit is computed and is compared with received CRC. The Myrinet switches employ exactly the same blocking-cut-through (wormhole) routing[6] used in MPP systems. The worst-case latency through a switch is 550ns. Figure 3 shows the organization of a Myrinet host interface and the internal details of the custom RISC LANai chip (version 7.2) on which interface is based. The interface consists of a PCI bridge that can adapt to a either 32 or 64 bit PCI bus, a DMA controller and a LANai 7 chip.

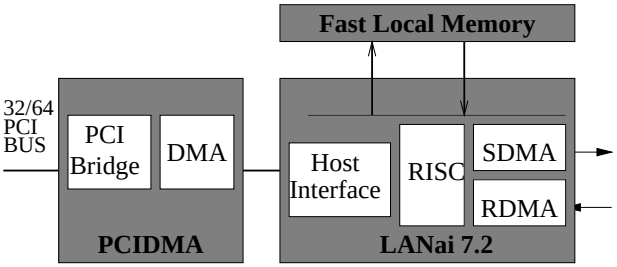

Figure 3. Myrinet PCI card.

A LANai chip contains a core RISC processor clocked at 66MHz, the Myrinet packet interface, a pipelined host interface (EBUS interface), and an interface (LBUS) to synchronous static SRAMS of 2MB. There are three DMA engines on a Myrinet interface card: two for data transfers between the network link and the SRAM, and one for moving data between the SRAM and host main memory over the PCI bus. The LBUS operates at twice the chip-clock speed (two LBUS memory cycles for every clock cycle) at 132MHz, enabling up to two DMA engines to operate at full speed at the same time. The host can access both the SRAM and the network itself, by using Programmed I/O (PIO) instructions. On the contrary, the LANai can only access host memory through host-to-LANai DMA engine.

3 Overview of user-level communication systems

Two user level communication systems are evaluated on the test configuration presented above. The two systems are GM 1.1.3 from Myricom and BIP 0.99 from University of Leon. These two systems, like other user level communication systems[7], share many common features such as direct user access to the network interface and minimal data copying. However, there are differences between these systems as well. The GM is a general message communication system that enables protected memory access to network in-

terface, supports both send/reply and a direct deposit style communication model, provides reliable and ordered delivery of messages and has internal flow control mechanism. In contrast BIP is a minimal library that aims at providing raw hardware performance to applications. It only supports a send/reply communication model, assumes reliable network and provides no flow control and error recovery. In addition GM+ uses PIO instead of DMA to pass small messages from user space to a Myrinet card, and offers an option to do unreliable delivery of small messages.

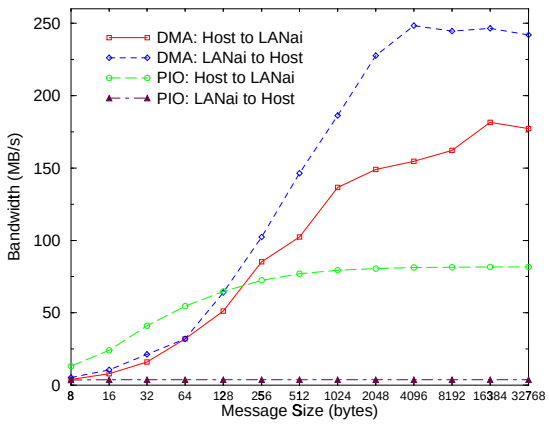

Figure 4. Performance of DMA and PIO.

4 Performance Evaluation

4.1 DMA versus PIO

There are two basic mechanisms to transfer bytes between host memory and a network interface card (NIC): Direct Memory Access (DMA) and Programmed I/O (PIO). With DMA, the NIC directly reads and writes the host's memory without any host CPU involvement; the host simply gives the NIC a memory address and the NIC reads from (writes to) it. With PIO, the host CPU is directly responsible for moving data between NIC and the host memory: to send a packet, the CPU sits in a tight loop that first reads a word from host memory and then writes it to the NIC; to receive a packet, the host CPU reads words from the NIC and then writes them to memory. Figure 4 shows the bandwidth of DMA and PIO for bytes transfer between host memory and the Myrinet NIC. DMA transfers are initiated by the LANai processor. It is clear that DMA bandwidth from LANai to host has higher bandwidth than DMA bandwidth from host to LANai for large messages while read PIO bandwidth is significant lower than write PIO bandwidth. These discrepancies are because read operations on PCI bus are more expensive than write operations. For small messages (up to 256 bytes), using PIO to transfer data from host to LANai has higher bandwidth because of the relatively high DMA initiation cost for small messages.

The user-level communication systems allow data transfer from arbitrary user address to NIC and vice versa. GM system sends user data through DMA but receives data through PIO for small messages (<= 256 bytes) instead. BIP system transfers small messages (<= 256 bytes) using PIO while it transfers large messages using DMA. Our enhancement to GM (GM+) sends small messages (<= 256 bytes) using PIO to avoid DMA initiation cost. All systems use two DMA engines to transfer data between the SRAM and links.

4.2 Bandwidth

There are three types of bandwidth that are important to applications. They are **Unidirectional bandwidth** which is the one way data transfer, **Bidirectional ping-pong bandwidth** which captures data flows in both direction alternatively, and **Bidirectional simultaneous bandwidth** which models data flow in both directions simultaneously.

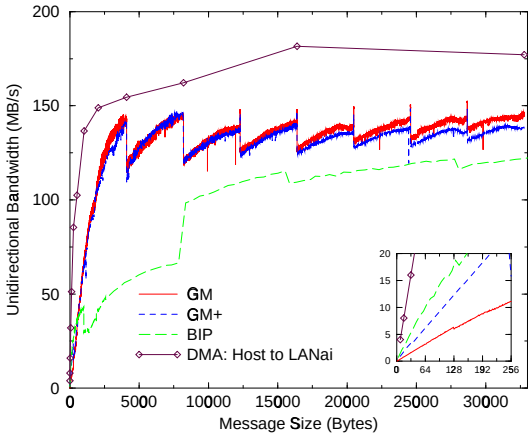

Figure 5. Unidirectional bandwidth.

The unidirectional bandwidth indicates how well data are transfered in one direction only. The sender does not wait for reply from receiver for each message. The upper bound of this type of bandwidth is governed by host to LANai DMA bandwidth. Figure 5 shows unidirectional bandwidth of GM, GM+ and BIP in addition to a plot of host to LANai DMA bandwidth. In the smaller message region where message length is smaller than 256 bytes, GM

has lower bandwidth than that of GM+ and BIP since GM+ and BIP use PIO to transfer small message instead of using DMA. This point is demonstrated again in BIP with a knee at 256 bytes where BIP is switching from PIO to DMA transfer. Beyond the small message region especially in the "usable region" (4k - 10k), GM and GM+ have much higher bandwidth than BIP since GM and GM+ utilize pipelined host to LANai DMA very well. Finally the reason for the "sawtooth" pattern on the graph in GM and GM+ is that GM fragments long messages into packets of at most 4KB at the sender, and reassembles the packets into messages at the receiver. This fragmentation and reassembly is performed in order to limit the packet size in the network, so that a long message will not block a channel for an extended period, but will allow other packets to be interleaved on the channel.

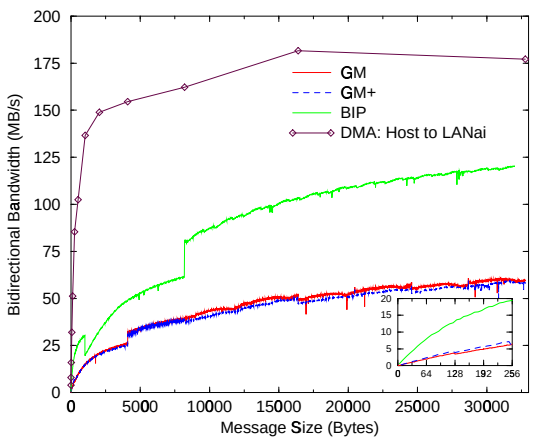

Figure 6. Bidirectional bandwidth.

The bidirectional pingpong bandwidth is resulted from situation in which data flows in both direction alternatively. A data sender waits after sending a message until it receives one. Figure 6 presents this type of bandwidth. It is clear that BIP has higher bandwidth than GM and GM+ have. This comes as no surprise since GM and GM+ support reliable data delivery by sending an acknowledgment upon receiving a message.

In bidirectional simultaneous bandwidth, data flows in both direction simultaneously. The data senders continuously send and receive data. This type of data transfer can take advantage of all six DMA engines of two network interfaces. Figure 7 summarizes the performance of this type of bandwidth.

The communication systems all use DMA to transfer large messages from user space to LANai space to avoid data copying. BIP and GM+ use PIO to transfer small messages to achieve better bandwidth in this region. GM utilizes pipelined DMA features very well and shows highest unidirectional bandwidth. However, the overhead of general APIs and reliable delivery of messages reduces GM bandwidth in the other two types of bandwidth.

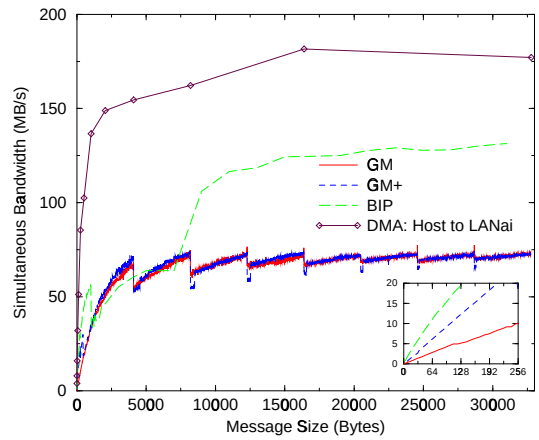

Figure 7. Bidirectional simultaneous bandwidth.

4.3 User to User Latency

This performance benchmark measures the time for a packet of arbitrary size to be sent from a user application on a node across the network to another application on a different node. The purpose of this measurement is to simulate communication that would occur between two applications running on different nodes in a cluster in which the receiver always sends a reply to the sender. This performance metric is commonly called the half Round Trip Time (RTT/2) or one way latency. The measurements are presented in Figure 8. It is clear that BIP, which offers the minimal functionalities, achieves the best latency for small messages of around 6 μs. Furthermore by comparison latency results between GM and GM+, it reveals that about 2 μs reduction of latency can be achieved by going from DMA to PIO and removal of reliable delivery mechanism for small messages.

4.4 LogP Metrics

In order to better understand the performance results presented above, we look at a simple cross section of the latency for sending a single packet. The time measured are derived from the LogP conceptual model which measures the following parameters of a communication systems: the time a message spends between the source and destination network interface (L), the time the host processor is

involved in sending or receiving a message (O_s, O_r), and the interval (g) between successive sends. Figure 9 shows the general pseudo-code of LogP micro-benchmarks. Code fragment (a) measures RTT/2 for all systems, code fragment (b) is used to measure O_s and g in all systems, and O_r in GM, and code fragment (c) measures O_r in BIP and GM+ (small messages). L is calculated from $\frac{RTT}{2} = O_s + O_r + L$. The spin(D) in the code segment spins for D μs. The send and recv calls are non-blocking if possible to achieve best performance.

Table 1 summarizes the LogP parameters for the communication systems with message length of 8 bytes. The gap g which is the minimum interval on the sending processor between successive sends at steady state is measured using part (b) of Figure 9 with zero D. The large gap of GM system is due to the fact that GM has limited send buffer and each send requires an acknowledgment from a receiver. The send overhead O_s is the host overhead of sending a message. For GM, O_s is measured using the part (b) of the Figure 9 when D is zero and N is below network buffering capacity. For the others, we use large D and the equation $g(D) = O_s + D$. The relative large result of O_s for BIP is caused by synchronous sends, which add PIO cost to value O_s. The receive overhead O_r models a host overhead of receiving a message. In GM, O_r is measured with code segment (b) in Figure 9 using equation $g(D) = O_s + O_r + max(Idle, D)$ since every send requires an acknowledgment, where $Idle$ is obtained by $Idle = D'$ for all $0 <= D <= D'$, $g(D) = g(0)$. In BIP and GM+ O_r is computed using part (c) in Figure 9 with equation of $g(D) = O_r + D$ for large D. In BIP a message is transfered to user memory using DMA without host intervention therefore BIP has smallest O_r value. Finally L is computed

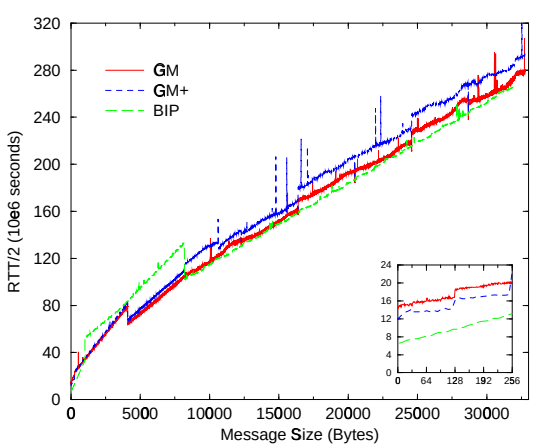

Figure 8. RTT/2 benchmark.

```
Barrier();                Barrier();
time();                   ;
for N times               for N times
  send();                   recv();
  recv();                   send();
time();                   ;
Barrier();                Barrier();
          (a) RTT/2
Barrier();                Barrier();
time();                   ;
for N times               for N times
  send();                   recv();
  spin(D);                  ;
time();                   ;
Barrier();                Barrier();
          (b) Os, Or and g
Barrier();                Barrier();
time();                   ;
for N times               for N times
  recv();                   send();
  spin(D);                  ;
time();                   ;
Barrier();                Barrier();
          (c) Or
Node 0                    Node 1
```

Figure 9. Pseudo-code for LogP parameters.

from the equation $\frac{RTT}{2} = O_s + O_r + L$, and it counts how much time a message is spent from one NIC to another NIC. Since BIP offers no reliable message delivery with minimal bookkeeping in its MCP code, it has smallest L value. The L value of GM+ is about 2.5μs less than that of GM since GM+ uses PIO to send message and offers no acknowledgment on receiving a message.

	$\frac{RTT}{2}$	Gap	O_s	O_r	L
GM	14.76	18.23	0.86	1.93	11.91
GM+	12.78	8.55	2.02	1.29	9.47
BIP	6.65	5.65	5.04	0.48	1.13

Table 1. LogP measurement results.

5 Related issues

The previous section summarizes performance results of the communication systems. These performance figures reveal some of most important issues such as reliable communication and data copying in achieving high bandwidth and low latency communication systems. However,there are some issues which need to be discussed.

- Interrupt Handling: The cost of delivering interrupts is part of the problem in traditional implementations of TCP protocol stack[8]. In order to avoid using expensive interrupts to deliver messages through the OS,

the user-level communication systems use DMA to deposit the message directly into user memory, and receivers then poll the network events.

- Connection establishment: GM and BIP offer no explicit connection establishment. Applications simply build a message and send it to a correct destination. However the communication systems maintain connection information between each pair of hosts in the network.

In addition to the above, issues such as network management, thread or fork safety, connection inheritance[9], and congestion control are either ignored or given little considerations in the user-level communication systems. Some of the issues may be left to higher level software but they have been studied very little in user-level communication systems.

6 Conclusion

In this paper two user level communication systems are studied extensively on the same platform. The three types of bandwidth, user-to-user latency and $LogP$ results are presented. The performance benchmark reveals not only common practices among these systems but also differences in decisions in both the communication model and implementations. The studies show inverse relationship between functionality and small message latency, and quantify the cost of reliable message delivery. In addition the studies show that short message latency lies in a range of $5 - 15\mu$s, and maximum bandwidth lies in a range of $50 - 140$ MB/s. In the future, a multi-priority packet based communication system, which aims at very low latency for small messages of unreliable delivery, derived from GM will be developed to address problems in applications that requires very low latency in global operations.

References

[1] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W.-K. Su. Myrinet: A gigabit-per-second local area network. *IEEE Micro*, 15(1):29-36, Feb. 1995.

[2] R. Gillett, M. Collins, and D. Pimm. Overview of network memory channel for PCI. In *Proceedings of the IEEE Spring COMPCON '96*, Feb. 1996.

[3] See *http://www.giganet.com/products/index.htm.*

[4] A. Basu, V. Bush, W. Vogels, and T. von Eicken. U-net: A user-level network interface for parallel and distributed computing. In *Proceedings of 15th ACM symposium on Operating Systems Principle (SOSP)*, Copper Mountain, Colorado, Dec. 1995.

[5] D. Culler, L. Liu, R. P. Martin, and C. Yoshikawa. LogP performance assessment of fast network interfaces. *IEEE Micro*, 1996.

[6] C. L. Seitz, W.-K. Su. In *Proceedings of the University of Washington Symposium on Integrated Systems*, pp. 320-337, MIT Press, 1993.

[7] S. Araki, A. Bilas, C. Dubnicki, J. Edler, K. Konishi, and J. Philbin. In *Proceedings of The 1998 SC98 conference*. Orlando, Florida. Nov, 1998.

[8] A. Barak, I. Gilderman, I. Metrik. Performance of the Communication Layers of TCP/IP with the Myrinet Gigbit LAN. *Computer Communications*, Vol. 22, No. 11, Jul. 1999.

[9] T. Anderson, B. Bershad, E. Lazowska, and H. Levy. Scheduler activations: Effective kernel support for the user-level management of parallelism. In *Proceedings of the 13th Symposium on Operating Systems Principles*, pp. 95-109, Oct. 1991.