High Performance Protocols for Clusters of Commodity Workstations

P. Melas and E. J. Zaluska

Electronics and Computer Science
University of Southampton, U.K.
pm95r@ecs.soton.ac.uk
ejz@ecs.soton.ac.uk

Abstract. Over the last few years technological advances in microprocessor and network technology have improved dramatically the performance achieved in clusters of commodity workstations. Despite those impressive improvements the cost of communication processing is still high. Traditional layered structured network protocols fail to achieve high throughputs because they access data several times. Network protocols which avoid routing through the kernel can remove this limit on communication performance and support very high transmission speeds which are comparable to the proprietary interconnection found in Massively Parallel Processors.

1 Introduction

Microprocessors have improved their performance dramatically over the past few years. Network technology has over the same period achieved impressive performance improvements in both Local Area Networks (LAN) and Wide Area Networks (WAN) (e.g. Myrinet and ATM) and this trend is expected to continue for the next several years. The integration of high-performance microprocessors into low-cost computing systems, the availability of high-performance LANs and the availability of common programming environments such as MPI have enabled Networks Of Workstations (NOWs) to emerge as a cost-effective alternative to Massively Parallel Processors (MPPs) providing a parallel processing environment that can deliver high performance for many practical applications.

Despite those improvements in hardware performance, applications on NOWs often fail to observe the expected performance speed-up. This is largely because the communication software overhead is several orders of magnitude larger than the hardware overhead [2]. Network protocols and the Operating System (OS) frequently introduce a serious communication bottleneck, (e.g. traditional UNIX layering architectures, TCP/IP, interaction with the kernel, etc). New software protocols with less interaction between the OS and the applications are required to deliver low-latency communication and exploit the bandwidth of the underlying intercommunication network [9].

Applications or libraries that support new protocols such as BIP [4], Active Messages [6], Fast Messages [2], U-Net [9] and network devices for ATM and

Myrinet networks demonstrate a remarkable improvement in both latency and bandwidth, which are frequently directly comparable to MPP communication performance.

This report investigates commodity "parallel systems", taking into account the internode communication network in conjunction with the communication protocol software. Latency and bandwidth tests have been run on various MPPs (SP2, CS2) and a number of clusters using different workstation architectures (SPARC, i86, Alpha), LANs (10 Mbit/s Ethernet, 100 Mbit/s Ethernet, Myrinet) and OS (Solaris, NT, Linux). The work is organised as follows: a review of the efficiency of the existing communication protocols and in particular TCP/IP from the NOW perspective is presented in the first section. A latency and bandwidth test of various clusters and an analysis of the results is presented next. Finally an example specialised protocol (BIP) is discussed followed by conclusions.

2 Clustering with TCP/IP over LANs

The task of a communication sub-system is to transfer data transparently from one application to another application, which could reside on another node. TCP/IP is currently the most widely-used network protocol in LANs and NOWs, providing a connection-oriented reliable byte stream service [8]. Originally TCP/IP was designed for WANs with relatively high error rate. The philosophy behind the TCP/IP model was to provide reliable communication between autonomous machines rather than a common resource [6].

In order to ensure reliability over an unreliable subnetwork TCP/IP uses features such as an in-packet end-to-end checksum, "SO_NDELAY" flag, "Time-to-live" IP field, packet fragmentation/reassembly, etc. All these features are useful in WANs but in LANs they represent redundancy and consume vital computational power [6].

Several years ago the bandwidth of the main memory and the disk I/O of a typical workstation was an order of magnitude faster than the physical network bandwidth. The difference in magnitude was invariably sufficient for the existing OS and communication protocol stacks to saturate network channels such as 10 Mbit/s Ethernet [3]. Despite hardware improvements workstation memory access time and internal I/O bus bandwidth in a workstation have not increased significantly during this period (the main improvements in performance have come from caches and a better understanding of how compilers can exploit the potential of caches). Faster networks have thus resulted in the gap between the network bandwidth and the internal computer resources being considerably reduced [3].

The fundamental design objective of traditional OS and protocol stacks at that time was reliability, programmability and process protection over an unreliable network. In a conventional implementation of TCP/IP a send operation involves the following stages of moving data: data from the application buffer are copied to the kernel buffer, then packet forming and calculation of the headers and the checksum takes place and finally packets are copied into the network

interface for transmission. This requires extra context switching between applications and the kernel for each system call, additional copies between buffers and address spaces, and increased computational overhead [7].

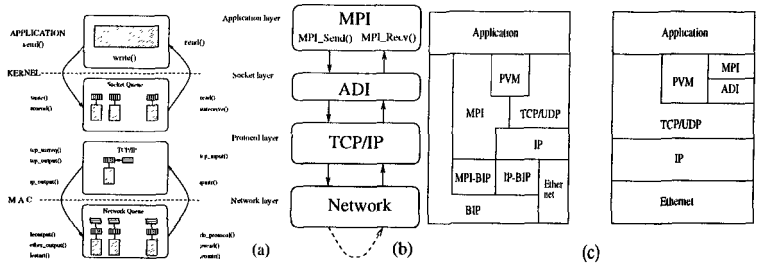

Fig. 1. a) Conventional TCP/IP implementation b)MPI on top of the TCP/IP protocol stack c)The BIP protocol stack approach

As a result, clusters of powerful workstations still suffer a degradation in performance even when a fast interconnection network is provided. In addition, the network protocols in common use are unable to exploit fully all of the hardware capability resulting in low bandwidth and high end-to-end latency. Parallel applications running on top of communication libraries (e.g. MPI, PVM, etc.) add an extra layer on top of the network communication stack, (see Fig. 1).

Improved protocols attempt to minimise the critical communication path of application-kernel-network device [4, 6, 2, 9]. Additionally these protocols exploit advanced hardware capabilities (e.g. network devices with enhanced DMA engines and co-processors) by moving as much as possible of their functionality into hardware. Applications can also interact directly with the network interface avoiding any system calls or kernel interaction. In this way processing overhead is considerably reduced improving both latency and throughput.

3 Latency and Bandwidth Measurements

In order to estimate and evaluate the effect of different communication protocols on clusters a ping-pong test measuring both peer-to-peer latency and bandwidth between two processors over MPI was written [2]. The latency test measures the time a node needs to send a sequence of messages and receive back the echo. The bandwidth test measures the time required for a sequence of back-to-back messages to be sent from one node to another. In both tests receive operations were posted before the send ones. The message size in tests always refers to the payload. Each test is repeated many times in order to reduce any clock jitter, first-time and warm-up effects and the median time from each measurement test presented in the results.

Various communication models [1, 5] have been developed in order to evaluate communication among processors in parallel systems. In this paper we follow a

linear approach with a start-up time α (constant per segment cost) and a variable per-byte cost β. The message length at which half the maximum bandwidth is achieved ($n_{1/2}$) is an important indication as well.

$$t_n = \alpha + \beta n \tag{1}$$

Parameters that can influence tests and measurements are also taken into account for each platform in order to analyse the results better, i.e. measurements on non-dedicated clusters have also been made to assess the effect of interference with other workload.

Tests on MPPs (SP2 and CS2) Latency and bandwidth tests have been run on the SP2 and CS2 in the University of Southampton and on the SP2 at Argonne National Laboratory. The results of the MPP test are used as a benchmark to analyse and compare the performance of NOW clusters. In both SP2 machines the native IBM's MPI implementation was used while on the CS2 the mpich 1.0.12 version of MPI was used. The difference in performance between the two SP2 machines is due to the different type of the high-performance switch (TB2/TB3). The breakpoint at 4 KB on the SP2 graph is due to the change in protocol used for sending small and large messages in the MPI implementation.

The NT cluster. The NT cluster is composed of 8 DEC Alpha based workstations and uses a dedicated interconnection network based around a 100Mbit/s Ethernet switch. The Abstract Device Interface makes use of the TCP/IP protocol stack that NT provides. Results on latency and bandwidth are not very impressive (Fig. 2), because of the experimental version of the MPI implementation used. The system was unable to make any effective use of the communication channel, even on large messages. An unnecessarily large number of context switches resulted in very low performance.

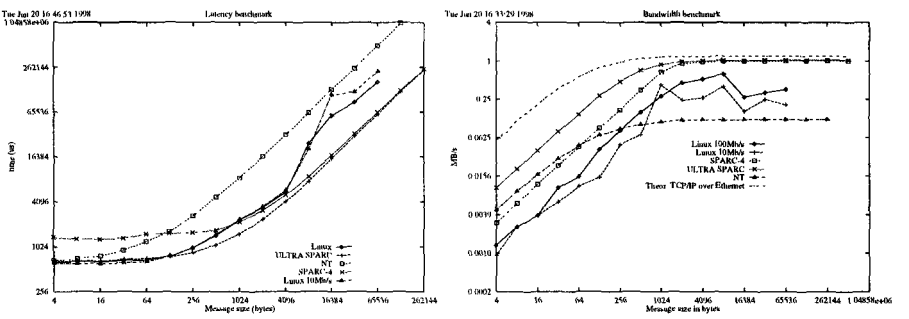

Fig. 2. Latency and bandwidth on NOW clusters

The Linux cluster. This cluster is build around Pentium-Pro machines running Linux 2.0.x. Tests have run using both the 10Mbit/s channel and FastEthernet with MPI version mpich 1.1. A poor performance was observed similar to the NT cluster, although in this case the non-dedicated interconnection network affected some of the results. Although its performance is slightly better than the NT cluster, the system fails to utilise the underlying hardware efficiently. In Fig.2 we can see a break point at the bandwidth plots close to 8 KB due to the interaction between TCP/IP and the OS.

The Solaris cluster. The Solaris cluster uses a non-dedicated 10Mbit/s Ethernet segment running SunOS 5.5.1 and the communication library is mpich 1.0.12. This cluster has the best performance among the clusters for 10Mbit/s Ethernet. Both SPARC-4 and ULTRA SPARC nodes easily saturated the network channel and push the communication bottleneck down to the Ethernet board. The non-dedicated intercommunication network had almost no affect on the results.

The Myrinet cluster. This cluster is build around Pentium-Pro machines running Linux 2.0.1, with an interconnection based on a Myrinet network. The communication protocol is the *Basic Interface for Parallelism* (BIP), an interface for network communication which is targeted towards message-passing parallel computing. This interface has been designed to deliver to the application layer the maximum performance achievable by the hardware [3]. Basic features of this protocol are a zero-copy protocol, user application level direct interaction with the network board, system call elimination and exploitative use of memory bandwidth. BIP can easily interface with other protocol stacks and interfaces such as IP or APIs, (see Fig. 1).

The cluster is flexible enough to configure the API either as a typical TCP/IP stack running over 10Mbit/s Ethernet, TCP/IP over Myrinet or directly on top of BIP. In the first case the IP protocol stack on top of the Ethernet network provides similar performance to the Linux cluster discussed above, although its performance is comparable to the Sun cluster. Latency for zero-size message length has dropped to 290 μs and the bandwidth is close to 1 MB/s for messages larger than 1 KB.

Changing the physical network from Ethernet to Myrinet via the TCP/BIP protocol provides a significant performance improvement. Zero size message latency is 171 μs and the bandwidth reaches 18 MB/s, with a $n_{1/2}$ figure below 1.5KB. Further change to the configuration enables the application (MPI) to interact directly with the network interface through BIP. The performance improvement in this case is impressive pushing the network board to the design limits. Zero length message latency is 11 μs and the bandwidth exceeds 114 MB/s with a $n_{1/2}$ message size of 8 KB.

Figure 3 shows the latency and bandwith graphs for those protocol stack configurations. A noticeable discontinuity at message sizes of 256 bytes reveals the different semantics between short and long messages transmission modes.

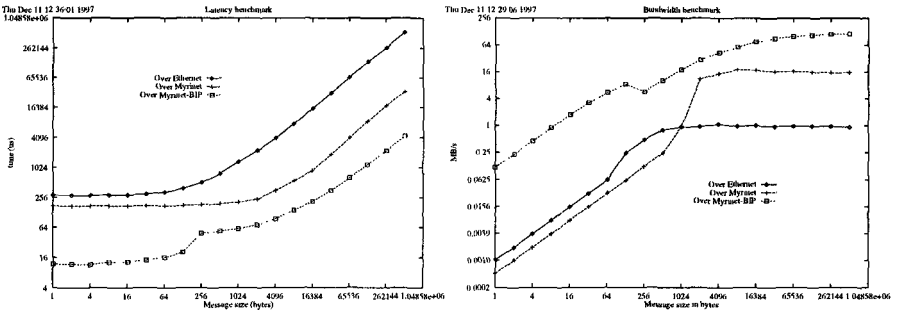

Fig. 3. Latency and bandwidth on a Myrinet cluster with page alignment

4 Analysis of the Results

The performance difference between normal NOW clusters and parallel systems is typically two or more orders of magnitude. A comparable performance difference can sometimes be observed between the raw performace of a NOW cluster and the performance delievered at the level of the user application (the relative latency and bandwidth performance of the above clusters is presented in Fig. 2). With reference to the theoretical bandwidth of a 10 Mbit/s Ethernet channel with TCP/IP headers we can see that only the Sun cluster approaches close to that maximum. The difference in computation power between the Ultra SPARC and SPARC-4 improves the latency and bandwidth of short messages.

On the other hand the NT and Linux cluster completely fail to saturate even a 10Mbit/s Ethernet channel and an order of magnitude improvement in the interconnection channel to FastEthernet does not improve throughput significantly. In both clusters the communication protocol implementation drastically limits performance. It is worth mentioning that both software and hardware for the Solaris cluster nodes comes from the same supplier, therefore the software is very well tuned and exploits all the features available in the hardware. By contrast the open market in PC-based systems requires software to be a more general and thus less efficient.

Similar relatively-low performance was measured on the Myrinet cluster using the TCP/IP protocol stack. The system exploits a small fraction of the network bandwidth and latency improvement is small. Replacing the communication protocol with BIP the Myrinet cluster is then able to exploit the network bandwidth and the application level performance becomes directly comparable with MPPs such as the SP2, T3D, CS2, etc.

As can be seen from Fig. 3, the Myrinet cluster when compared to those MPPs has significantly better latency features over the whole range of the measurements made. In bandwidth terms for short messages up to 256 bytes Myrinet outperforms all the other MPPs. Then for messages up to 4KB (which is the breakpoint of the SP2 at Argonne), the SP2 has a higher performance, but after this point performance of Myrinet is again better.

Table 1. Latency and Bandwidth results

Configuration	Cluster	H/.W	min Lat.	max BW	$n_{1/2}$
MPI over TCP/IP	NT/Alpha	FastEth.	673 μs	120 KB/s	100
MPI over TCP/IP	Linux/P.Pro	Ethernet	587 μs	265 KB/s	1.5 K
MPI over TCP/IP	Linux/P.Pro	FastEth.	637 μs	652 KB/s	1.5 K
MPI over TCP/IP	SPARC-4	Ethernet	660 μs	1.03 MB/s	280
MPI over TCP/IP	ULTRA	Ethernet	1.37 ms	1.01 MB/s	750
MPI over TCP/IP	Linux/P.Pro	Ethernet	280 μs	1 MB/s	300
MPI over TCP/BIP	Linux/P.Pro	Myrinet	171 μs	17.9 MB/s	1.5 K
MPI over BIP	Linux/P.Pro	Myrinet	11 μs	114 MB/s	8 K

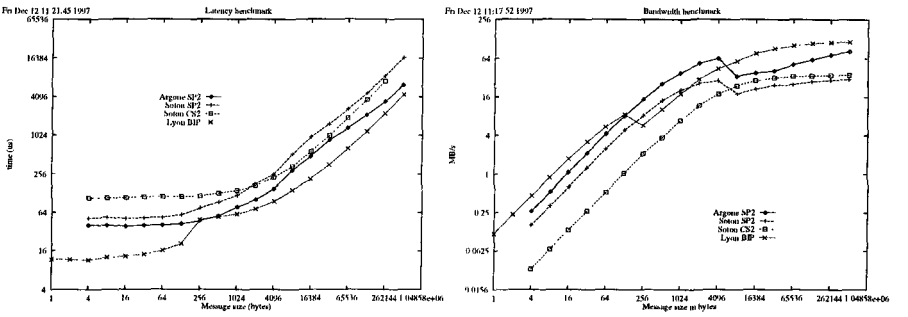

Fig. 4. Comparing latency and bandwidth between a Myrinet cluster and MPPs

5 Conclusion

In this paper the importance and the impact of some of the most common interconnecting network protocols for clusters has been examined. The TCP/IP protocol stack was tested with different cluster technologies and comparisons made with other specialised network protocols. Traditional communication protocols utilising the critical application-kernel-network path impose excessive communication processing cost and cannot exploit any advanced features in the hardware. Conversely communication protocols that enable applications to interact directly with network interfaces such as BIP, Fast Messages, Active Messages, etc can improve the performance of clusters considerable.

The majority of parallel applications use small-size messages to coordinate program execution and for this size of message latency overhead dominates the transmission time. In this case the communication cost cannot be hidden by any programming model or technique such as overlapping or pipelining. Fast communication protocols that deliver a drastically reduced communication latency are necessary for clusters to be used effectively in a wide range of scientific and commercial parallel applications as an alternative to MPPs.

ACKNOWLEDGEMENTS We acknowledge the support of the Greek State Scholarships Foundation, Southampton University Computer Services, Argonne National Laboratory and LHPC in Lyon.

References

1. J. J. Dongarra and T. Dunigan. Message-passing performance of various computers. Technical Report UT-CS-95-299, Department of Computer Science, University of Tennessee, July 1995.
2. Mario Lauria and Andrew Chien. MPI-FM: High performance MPI on workstation clusters. *Journal of Parallel and Distributed Computing*, 40(1):4–18, January 1997.
3. Bernard Tourancheau Loïc Prylli. Bpi: A new protocol design for high performance networking on myrinet. Technical report, LIP-ENS Lyon, September 1997.
4. Loic Prylli. Draft: Bip messages user manual for bip 0.92. Technical report, Laboratoire de l'Informatique du Parallelisme Lyon, 1997.
5. PARKBENCH Committee: Report-1, assembled by Roger Hockney (chairman), and Michael Berry secretary). Public international benchmarks for parallel computers. Technical Report UT-CS-93-213, Department of Computer Sci/ence, University of Tennessee, November 1993.
6. Steven H. Rodrigues, Thomas E. Anderson, and David E. Culler. High-performance local-area communication with fast sockets. In USENIX, editor, *1997 Annual Technical Conference, January 6–10, 1997. Anaheim, CA*, pages 257–274, Berkeley, CA, USA, January 1997. USENIX.
7. A. Chien S. Pakin, M. Lauria. High performance messaging on workstations: Illinois fast messages (fm) for myrinet. In *In Supercomputing '95*, San Diego, California, 1995.
8. W. R. Stevens. *TCP/IP Illustrated, Volume 1; The Protocols.* Addison Wesley, Reading, 1995.
9. Thorsten von Eicken, Anindya Basu, Vineet Buch, and Werner Vogels. U-net: a user-level network interface for parallel and distributed computing. In *Proceedings of the 15th ACM Symposium on Operating Systems Principles (SOSP)*, volume 29, pages 303–316, 1995.