Vol.16 No.6 J. Comput. Sci. & Technol. Nov. 2001

BCL-3: A High Performance Basic Communication Protocol for Commodity Superserver DAWNING-3000

MA Jie (马 捷), HE Jin (贺 劲), MENG Dan (孟 丹) and LI Guojie (李国杰)

Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, P.R. China

E-mail: {majie, jhe, md, lig}@ncic.ac.cn

Received December 28, 2000; revised April 11, 2001.

Abstract This paper introduces the design and implementation of BCL-3, a high performance low-level communication software running on a cluster of SMPs (CLUMPS) called DAWNING-3000. BCL-3 provides flexible and sufficient functionality to fulfill the communication requirements of fundamental system software developed for DAWNING-3000 while guaranteeing security, scalability, and reliability. Important features of BCL-3 are presented in the paper, including special support for SMP and heterogeneous network environment, semi-user-level communication, reliable and ordered data transfer and scalable flow control. The performance evaluation of BCL-3 over Myrinet is also given.

Keywords cluster communication system, user-level communication, scalable flow control, CLUMPS

1 Introduction

Clusters have been popular platforms for high performance computing in recent years. They are widely used in scientific and engineering computing business computing, and Internet information services. Communication performance is one of the most critical factors determining the performance of a whole cluster system. So how to improve the performance of communication is a hot research topic in cluster computing. Meanwhile, building a cluster with commodity SMPs (CLUMPS) is becoming a trend since this approach can increase the packing density of system while reducing cost. CLUMPS provides different hardware for intra- and inter-node communication, that is, shared memory and message-passing network. Communication software should take the difference into account and provide corresponding protocols so that the capacity of underlying communication hardware can be fully used for various applications[1].

BCL-3 is a high-performance low-level communication protocol designed for and implemented on DAWNING-3000. It provides reliable and ordered message passing with flow control and error correction. DAWNING-3000 is a CLUMPS consisting of 70 nodes. Each node is a 4-way 375MHz Power3 SMP, running IBM AIX4.3.3. All nodes are connected with Fast Ethernet, Myrinet and Dnet. Dnet is high performance interconnect developed on our own. Both Myrinet and Dnet are supported by BCL-3. It consists of a library in the user space, a device driver in the kernel space, and a control program called MCP running on the network interface. A few important features of BCL-3 are described as follows.

• Flexible and sufficient functionality

BCL-3 provides a channel-based asynchronous communication mechanism, which supports point-to-point message passing and remote memory access (RMA) operations. It provides efficient communication support for the implementations of high-level communication software and other system software on DAWNING-3000, including PVM (parallel virtual machine), MPI (message-passing interface), JIAJIA (a software DSM) and a cluster file system called COSMOS. BCL-3 also implements

This work is supported by the National '863' High-Tech Programme of China (No.863-306-ZD01-01).

automatic node status detection, node isolating/rejoining functionality, and application program exception handling to guarantee system availability and stability. Although more functionality can decrease communication performance, especially latency, it is necessary for a commodity system.

• Support for SMPs

BCL-3 provides different protocols for intra- and inter-node communication respectively. Intra-node communication is implemented via shared memory while inter-node via message-passing network. This improves intra-node communication performance dramatically.

• Zero memory copy and pin-down cache

✓

BCL-3 implements direct user-to-user data transfer within inter-node communication. Sending and receiving data buffers provided by users can be pinned down in physical memory and therefore become DMA-able during communication process. The pin-down Cache is implemented to reuse the pinned down area. All these make contributions to bandwidth improvement. BCL-3 provides almost all hardware bandwidth to its users.

• Semi-user-level communication

BCL-3 implements user-level communication on receiving side. But on sending side, there is a need for senders to trap into OS kernel and invoke message-sending operations. Compared with user-level communication, only one kernel trapping is incorporated into the communication path. Although it can increase communication latency a little bit, this combination has a few advantages, which will be discussed in later section.

• Heterogeneous network environment support

Because the BCL-3 library in user space is independent of underlying network interface, binary code written in it or in a high-level communication library such as PVM, MPI and JIAJIA on top of it can run on any combination of networks. Applications written in BCL-3 need not to be recompiled. This feature is especially useful for applications running over a cluster of clusters.

• Scalability and reliability

BCL-3 adopts ACK/NAK mechanism to guarantee reliable and ordered data communication. ACK/NAK-based flow control improves the scalability in comparison with credit-based flow control. Memory consumption is no longer proportional to system scale. Currently, ACK/NAK mechanism is implemented on the network interfaces of Myrinet and Dnet.

The communication performance of BCL-3 has been measured on DAWNING-3000 over Myrinet. As a reliable and ordered message-passing protocol, it provides 2.7μs one-way latency and 391MB/s bandwidth of intra-node communication while 18.3μs and 146MB/s of inter-node's. Performance of MPI on top of BCL-3 is also presented.

Sections 2 and 3 introduce the design and implementation of BCL-3 over Myrinet in detail. Then the evaluation of BCL-3 on DAWNING-3000 platform is shown in Section 4. Finally, we present our conclusions and discuss ongoing research in Section 5.

2 Design Issues

• Communication services

Services provided by a communication system can greatly impact its performance. Many communication protocols only provide limited services and show excellent performance. However, the limited services will make other system software on high layers more difficult to implement. BCL-3 provides reliable and ordered message passing with flow control and error correction. Although reliable and ordered delivery reduces the performance of BCL-3, it will reduce more performance when it is implemented on higher layers.

Furthermore, RMA and special message arrival informing mechanism are also provided to support DSMs and distributed file systems. RMA makes it possible to deliver messages without cooperation of the other side. It is essential for software DSMs and distributed file systems. Also, BCL-3 provides "select" mechanism to inform the arrival of messages so that the application can use BCL-3 message passing as a socket. It makes the applications using TCP/IP easy to be ported to using BCL-3.

• Semi-user-level communication

In traditional communication protocols, such as TCP/IP, the main communication path involves OS kernel trapping and interrupts handling. All these overheads cause high end-to-end message latencies. User-level communication allows applications to directly access the network interface cards (NIC) without operating system intervention on both sending and receiving sides. Messages are transferred directly to and from user-space by the network interface card while observing the traditional protection boundaries between processes. It reduces the communication overhead by avoiding kernel traps on the data path.

However, in commodity systems, security is the most important requirement. User level communication protocol exposes all the control data structures to user space so that any mistake or malice operation will cause the system to break down. Kernel-level communication protocol protects the important data structures in kernel space.

To provide a low-latency and secure communication, BCL-3 uses a combined method which ensures the system security by protecting the most important data structures in kernel space and reduces communication overhead by diminishing some unnecessary kernel traps. Furthermore, it can make the application binary portable, suitable in heterogeneous network environments and more efficient. BCL-3 is separated into two parts, the user-level library and the kernel extension. All the system-related operations are hidden in the kernel extension, so that the user-level library can be the same on different platforms. Application, which links the same user-level library, can be ported to different platforms with binary codes. Also, the differences between heterogeneous network environments are hidden in the kernel extension.

• Communication mechanism and semantics

There are three commonly used message-passing modes, synchronous message passing, blocking message passing and non-blocking message passing. It is clear that synchronous and blocking message passing can be implemented by non-blocking message passing. So BCL-3 only provides non-blocking sending and receiving semantics.

Channels are used to identify the message transfer paths in BCL-3. The system channel is used to transfer small messages. The rendezvous semantic is used to transfer large messages. Rendezvous means the send is guaranteed to complete only if a corresponding receive has been posted. RMA operations are presented in BCL-3 by using a special type of channel. A message arrival informing mechanism, which is compatible with the TCP/IP socket, is also provided so that applications can use the "select" system call to wait for the incoming messages.

• Reliability and scalability

Although the reliable protocol can be implemented at higher levels, it is more efficient to implement the reliable protocol in this communication layer. BCL-3 provides a reliable and ordered message passing data path. It uses an ACK/NAK based reliable protocol. By using CRC verification and sequential number checking on each packet transferred on the net, BCL-3 can ensure the correctness and order of packets. Since the communication hardware is highly reliable, this ACK/NAK based protocol is very efficient on our platform. Compared with credit-based reliable protocol, the ACK/NAK based protocol is scalable. It is easy for the protocol to be used in large-scale cluster.

• Zero memory copy and buffer management

To implement zero memory copy, the user buffer should be pinned in physical memory and the virtual address should be translated into physical address before starting communication. This operation will increase the communication overhead. In order to reduce the communication overhead, the pin-down cache[2] and three types of buffers are introduced in BCL-3. They can reduce the overhead and improve the system performance. These two techniques will be discussed in detail in the next section.

• Special support for SMPs

In the context of SMPs, communication protocol needs to support several processes on one node. When the memory copy bandwidth is lower than the network bandwidth, it will provide a high performance to transfer messages via NIC even if the two processes are in the same node. But the memory copy bandwidth is improved heavily now. It is costly and has no meaning to transfer intra-node messages as the same manner to transfer inter-node messages. BCL-3 uses direct memory copy instead of transfer via network. Shared memory and direct copy from another process's memory space can be used to implement intra-node message passing. BCL-3 uses the former mechanism, which will be discussed later[3–5].

3 BCL-3 Protocol

BCL-3 is a low-level communication software used on DAWNING-3000. It has two versions. One is implemented on the Myrinet and the other is implemented on the Dnet. The upper levels can make full use of the hardware performance via BCL-3. BCL-3 supports point-to-point message passing. All other collective message passing should be implemented in the higher level software.

3.1 Architecture

Fig.1 is the protocol stack from the angle of the BCL-3 applications. Applications can directly access the BCL-3 level or use other functionality levels. BCL-3 is the lowest level software in DAWNING-3000's communication software. MPI (Message-Passing Interface) is implemented directly on BCL-3. PVM (Parallel Virtual Machine) is implemented on ADI-2 (Abstract Device Interface) of MPI. Notice that this may be changed according to different implementations that are done. In order to keep high performance, we can also port PVM directly on top of BCL-3. TCP/IP is designed to port onto our BCL-3 in the next step. A prototype will be completed early next year. The other two components in the stack are JIAJIA and COSMOS. The former is a DSM (Distributed Shared Memory) software, and the latter is a distributed file system of DAWNING-3000.

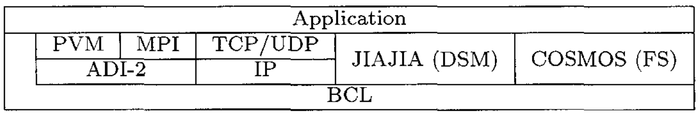

Fig.1. Protocol stack of DAWNING-3000 software.

BCL-3 is divided into three parts, the on-card control program (MCP, Myrinet Control Program), the device driver (DD) and the BCL library. Myrinet has an on-card processor, which can control the card to transfer packets by using three DMA engines. In BCL-3, MCP controls all the inter-node packet transfers. MCP completes a sending operation by reading send request in the card's local memory, sending/receiving messages with DMA engines and informing the user process of completion. Device driver is a kernel extension, which provides several *ioctl*() calls to control the hardware. It posts operation requests to the on-card memory. These requests include the sending requests and other requests, such as initialize/close communication port requests and initialize channel requests. Device driver also implements some functional operations, which need to be executed in the kernel environment. Such operations include the memory pin/unpin operation and physical memory address conversion. BCL library includes all implementations of BCL API.

3.2 Communication Mechanism

Communication is occurred between ports. Each process can create only one port to communicate with others. A process is labeled with its node number and port number. The pair of node number and port number is the unique identifier of the process within the application. Each port has a sending request queue, a receiving buffer pool and the corresponding event queues (sending event queue and receiving event queue). The receiving buffer pool is composed of several channels. There are three types of channels, system channel, normal channel and open channel.

When a process (sender) wants to send a message to another process (receiver), the sender should compose a send request and put it into the send request queue. Destination is specified by its node number, port number and channel number in the request. The receiving channel should be ready before the message arrives at the receive side. A receive event will be generated when a complete message arrives. On the send side, a send event will be put into the send event queue when a sending operation is over.

As shown in Fig.2, the system channel is designed to transfer small messages. Each process has one system channel. Every system channel has a buffer pool, which is initialized when the process starts. It is organized as an FIFO queue. When a short message arrives, it will automatically be put to the first free buffer in the buffer pool. The incoming message will be discarded if there is no free buffer in the pool. After the receiver gets the message, the buffer will be returned to the buffer pool.

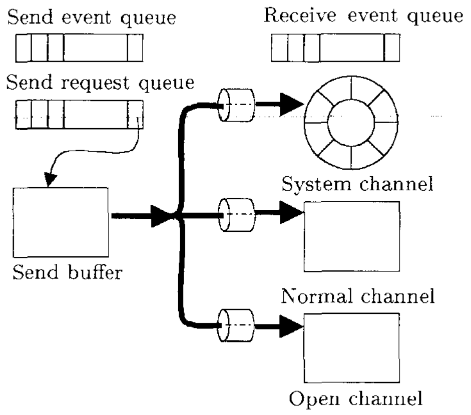

Fig.2. Communication mechanism of BCL-3.

The normal channel is designed to transfer messages with the rendezvous semantics. Each process has several normal channels. A normal channel has a user-specified buffer. The receiving process needs to prepare the receive channel before the sending process starts transmission. A receive event will be generated when a message completed. A normal channel should be initialized before each receiving operation.

The open channel is designed to perform RMA. Each process has several open channels. An open channel has a user-specified buffer. Unlike the normal channel, only once does it need to be initialized before receiving operations. A receiving event will be generated when an RMA operation is completed.

3.3 Buffer Management

A zero copy message transfer is realized with the help of the Myrinet DMA capability. The user program accesses only virtual addresses while the DMA engines access only physical memory addresses. To realize the DMA-issued message transfer, both the sending and receiving buffers must be pinned in the physical memory and the physical memory addresses must be known. The pin-down operation is costly in most operating systems. This overhead can be reduced if an application repeatedly transfers data from the same memory area without releasing the pinned-down area. When a request to release a pinned-down memory area is issued, the actual release operation is postponed until total physical memory reserved for the pinned-down area is used up. If the release primitive for pinned-down area has been issued but in fact the area remains pinned down, a subsequent request to pin down the same area can be answered immediately. This technique is called pin-down cache.

While using the pin-down cache technique, there is still a cache-hit overhead. BCL-3 avoids it by using the pinned buffer in message passing. If an application needs to transfer data from the same memory area repeatedly, it can request to pin down the buffer before transfer. The pinned-down buffer should be released after all transmissions. BCL-3 defines three types of buffer: normal buffer, enabled buffer and DMA buffer. A normal buffer is a user buffer that is not pinned. The communication primitives need to perform pin/unpin operation when using this type of buffer. An enabled buffer is a user buffer that is already pinned in the physical memory. A DMA buffer is a system buffer allocated by a special call, which is pinned in the physical memory. There is no need to check the pin-down cache before transmission when using the last two types of buffers. It reduces the overhead when using the pin-down cache.

3.4 Intra-Node Communication

There are several ways to move data from one process to another within one node (Fig.3). The traditional way is to move data as the same way as between nodes. Process A first transfers the data to NIC (Network Interface Circuit) by DMA. Then NIC transfers them back to process B. While the memory copy bandwidth is much higher than the DMA bandwidth, a good solution is to use shared memory to implement intra-node communication. Process A first copies the data to a shared memory area. Then process B copies them out. Another way is to move data directly from user space to user space.

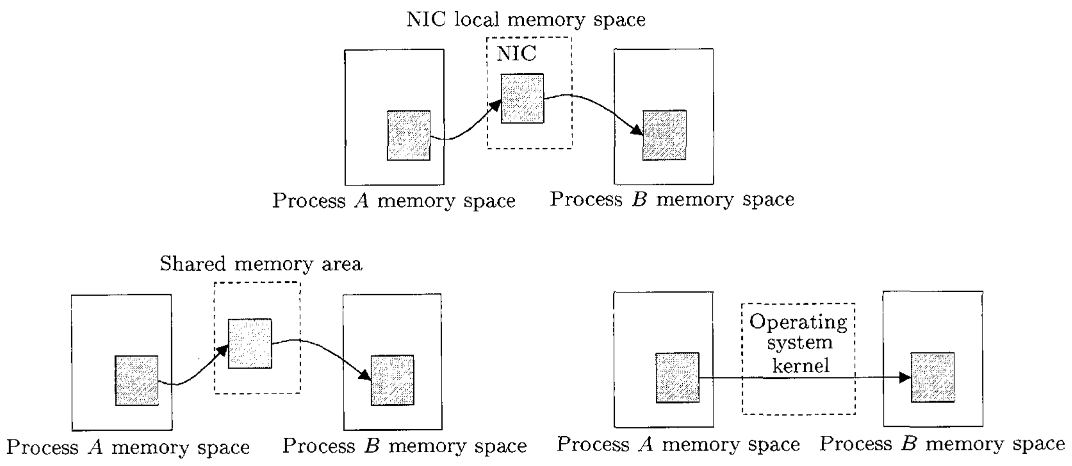

Fig.3. Several ways to move data between processes within one node.

Shared memory based intra-node communication can make the system more reliable. Any mistake or malice operation during a directly inter-process memory access can cause the target process to crash. By using shared memory, the sender process can only destroy the shared area. It will not affect other processes' space. This guarantees the independence of each process.

BCL-3 uses shared memory based intra-node communication. The internal buffer queue is used to transfer messages from one process to another within a node. This queue consists of a list of buffers. Each pair of processes has two queues (Fig.4). To ensure the message sequence, BCL-3 uses the sequential number to decide whether the operation should continue or not. Shared memory based intra-node communication needs an extra memory copy other than the direct memory copy solution. BCL-3 reduces the extra overhead by using the pipeline message passing technique.

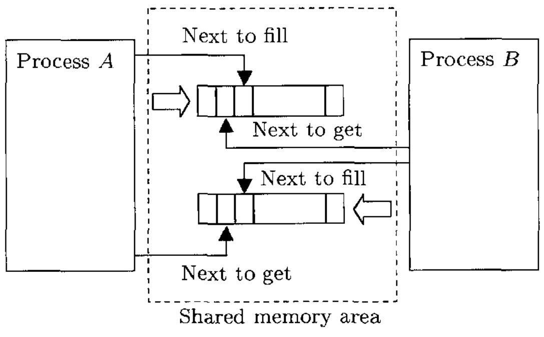

Fig.4. Intra-node communication in BCL-3.

3.5 Flow Control

BCL-3 provides reliable and ordered delivery. The receiving process will simply discard the packet when it detects an error. The sending process will retransmit it when timeout. When a source node sends two messages to the same destination, a receiving operation executed firstly at the destination will receive the first message. A sequential number based ACK/NAK protocol is used in BCL-3 to implement reliable and ordered delivery. Considering the high reliability of the network hardware, this algorithm is efficient.

Flow control is used to avoid buffer overflow and system deadlock. In BCL-3, the receiving process simply discards messages when there is no enough buffers. The sending process will re-send them when timeout. An overflowed packet is handled as an error packet.

4 Performance and Analysis

All the tests[6] are done on DAWNING-3000, which consists of 70 IBM270 workstations. Each node is a 4-way 375MHz Power3 SMP, which is running IBM AIX4.3.3. Myrinet[7] M2M-PCI64A NICs are used on each node. All nodes are interconnected by M2M-OCT-SW8 switches.

The first tests are raw point-to-point communications. Latency and bandwidth are measured on DAWNING-3000. Both the inter-node and the intra-node communications are tested. The result is shown in Fig.5 and Fig.6. The minimal latency is 2.7μs within one node and 18.3μs between nodes. The bandwidth is 391MB/s within one node (with the effect of cache) and 146MB/s between nodes. And the half-bandwidth is reached with less than 4KB message.

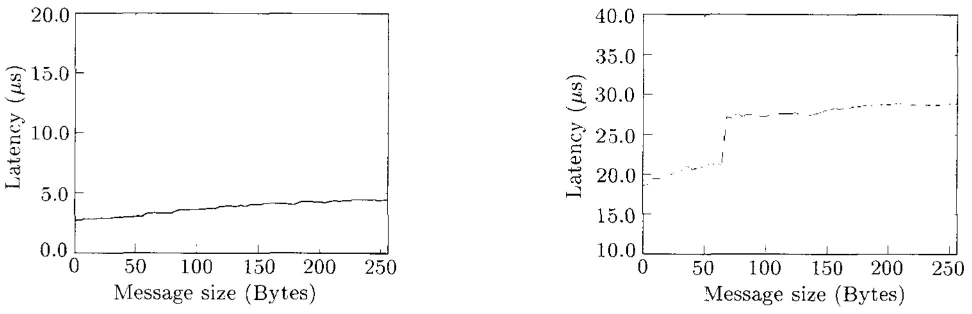

Fig.5. Intra-node (left) and inter-node (right) latency of BCL-3.

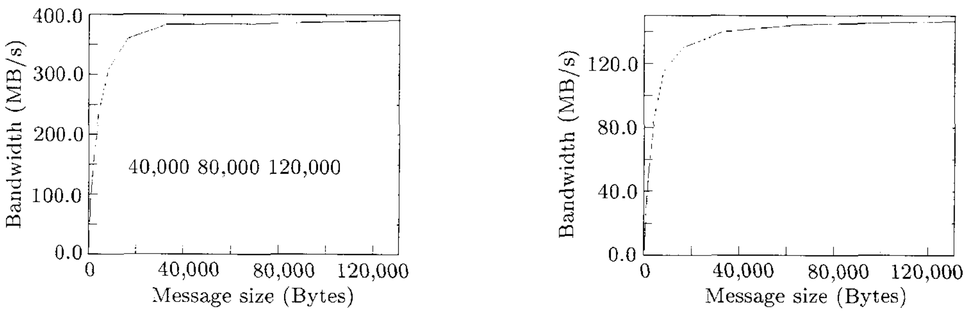

Fig.6. Intra-node (left) and inter-node (right) bandwidth of BCL-3.

Table 1 shows the comparison of three protocols. GM is a message-based communication system for the Myrinet, which is designed and implemented by Myricom. GM does not provide special support for SMP. Only inter-node communication performance data are given in the table. The range for GM's one-way short-message latency on a wide variety of hosts is from 13.37 to 21μs. And the peak bandwidth is over 140MB/s. BCL-3 reaches almost the same performance and provides a more reliable (using the kernel-level communication) and more complex (special support for SMP) protocol.

Table 1. Comparison of Different Communication Protocols

Protocol	Latency (μs)		Bandwidth (MB/s)	
	Intra-node	Inter-node	Intra-node	Inter-node
GM	–	13.37–21	–	>140
AM-II	3.6	27.5	160	32.8
BIP-SMP	1.8	5.7	160	126
BCL-3	2.7	18.3	391	146

AM-II[8–11] (Active Messages) is similar to remote procedure call (RPC) mechanism. Compared with AM-II, BCL-3 has a better latency in both the intra-node and the inter-node communications. It is meaningless to compare the bandwidths of these two protocols since AM-II needs an extra memory copy when transferring a message while BCL-3 does not. BCL-3 reaches a much higher bandwidth.

BIP[12–14] (Basic Interface for Parallelism) is a low level message passing system developed by the Laboratory for High Performance Computing in Lyon, France. It has a very low latency. But it does not provide the functionality of flow control and error correction. Its bandwidth is lower than that of BCL-3.

Fig.7 and Fig.8 show the performance of MPI over BCL-3. The minimal latency is 6.3μs within one node and 23.7μs between nodes. The bandwidth is 328MB/s within one node (with the effect of cache) and 131MB/s between nodes. When the message length grows to 8MB, the intra-node decreases. This shows the influence of cache. The bandwidth can be very high without cache replacement.

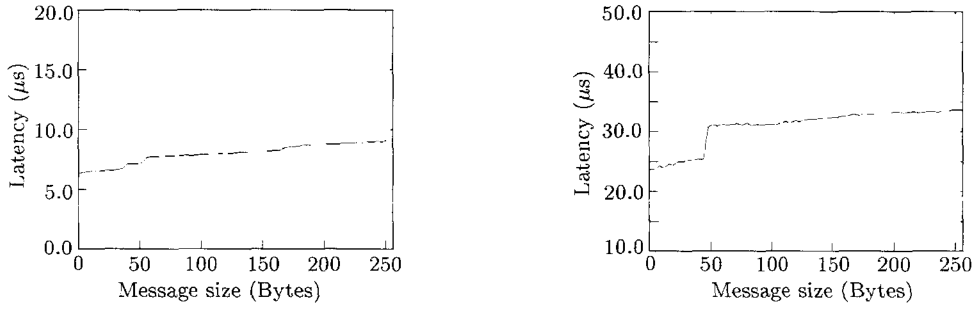

Fig.7. Intra-node (left) and inter-node (right) latency of MPI over BCL-3.

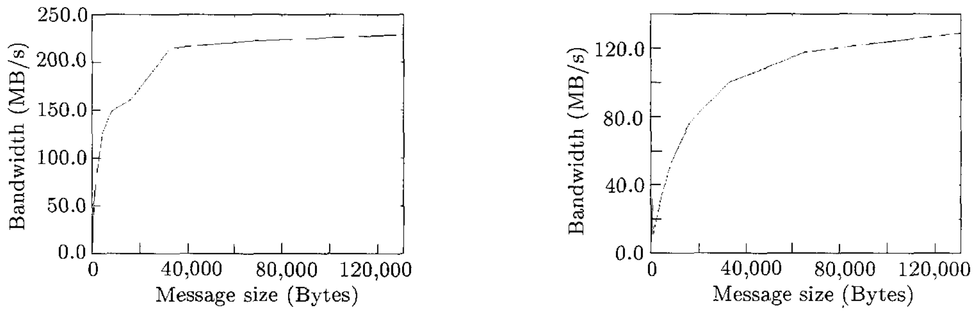

Fig.8. Intra-node (left) and inter-node (right) bandwidth of MPI over BCL-3.

5 Conclusion and Future Work

BCL-3 is available to easily build clusters of commodity SMPs. This communication layer combined with MPI over BCL-3 provides an efficient message passing interface for a cluster of commodity SMPs. BCL-3 achieves 2.7μs of latency and 391MB/s of bandwidth for intra-node message-passing. The performance of inter-node communication is 18.3μs of latency and 146MB/s of bandwidth. MPI over BCL-3 can achieve 6.3μs of latency and 328MB/s of bandwidth for intra-node message passing, and 23.7μs of latency and 131MB/s of bandwidth for inter-node message passing. BCL-3 is a flexible and reliable protocol. Point-to-point message passing and RMA are both presented in BCL-3. The combination of the kernel and user-level communications makes the system reliable and portable.

Although BCL-3 is at this time implemented on AIX, it can be ported to other platforms easily because it does not need to modify the source codes of the operating system kernel.

The bandwidth provided by Myrinet is approximately 160MB/s (1.28Gbit/s). BCL-3's bandwidth almost reaches the hardware limitation. To break the limitation, we plan to design a double-card system that uses two Myrinet cards to perform communication. It ought to get a higher bandwidth by this means.

Finally, we have tested BCL-3 on a large-scale cluster of SMP nodes, DAWNING-3000. There are 70 nodes and each node has four processors. This shows the scalability of BCL-3.

References

[1] Hwang Kai, Xu Zhiwei. Scalable Parallel Computing: Technology, Architecture, Programming. WCB/McGraw-Hill, 1998.

[2] Hiroshi Tezuka, Francis O'Carroll, Atsushi Hori, Yutaka Ishikawa. Pin-down cache: A virtual memory management technique for zero-copy communication. In *Proc. 12th International Parallel Processing Symposium & 9th Symposium on Parallel and Distributed Processing*. Florida, USA, 1998, pp.308–314.

[3] Foster I, Geisler J, Kesselman C, Tuecke S. Managing multiple communication methods in high-performance networked computing systems. *Journal of Parallel and Distributed Computing*, 1997, 40: 35–48.

[4] Gropp W W, Lusk E L. A taxonomy of programming models for symmetric multiprocessors and SMP clusters. In *Proc. Programming Models for Massively Parallel Computers*, Berlin, Germany, 1995, pp.2–7.

[5] Mellor-Crummey J M, Scott M L. Algorithms for scalable synchronization on shared-memory multiprocessors. *ACM Transactions on Computer Systems*, 1991, 9(1): 21–65.

[6] Luecke G R, Coyle J J. Comparing the performance of MPI on the Cray T3E-900, the Cray origin 2000 and the IBM P2SC. *Electronic Journal of Performance Evaluation and Modelling for Computer Systems (PEMCS)*, 1998, available at URL: http://hpc-journals.ecs.soton.ac.uk/PEMCS/.

[7] Boden N J, Cohen D, Felderman R E, Kulawik A E, Seitz C L, Seizovic J N, Su W. Myrinet — A gigabit-per-second local-area network. *IEEE Micro*, 1995, 15(1): 29–38.

[8] T von Eicken, Culler D E, Goldstein S C, Schauser K E. Active messages: A mechanism for integrated communication and computation. In *Proc. the 19th Annual International Symposium on Computer Architecture*, Gold Coast, Gueensland, Australia, 1992, pp.256–266.

[9] T von Eicken, Avula V, Basu A, Buch A. Low-latency communication over ATM networks using active messages. *IEEE Micro*, 1995, 15(1): 46–53.

[10] Lumetta S S, Mainwaring A M, Culler D E. Multi-protocol active messages on a cluster of SMP's. In *Proc. Supercomputing 97*, California, USA, 1997, electronic proceedings only, available at URL: http://www.supercomp.org/sc97/proceedings/.

[11] Lumetta S S, Culler D E. Managing concurrent access for shared memory active messages. In *Proc. the International Parallel Processing Symposium*, Florida, USA, 1998, pp.272–278.

[12] Prylli L, Tourancheau B. BIP: A new protocol designed for high-performance networking on Myrinet. In *Workshop PC-NOW, IPPS/SPDP98*, Orlando, USA, 1998, pp.472–485.

[13] Patrick Geoffray, Loïc Prylli, Bernard Tourancheau. BIP-SMP: High performance message passing over a cluster of commodity SMP's. In *Proc. Supercomputing 99*, 1999, electronic proceedings only, available at URL: http://www.supercomp.org/sc99/proceedings/.

[14] Loïc Prylli, Bernard Tourancheau, Roland Westrelin. An improved NIC program for high-performance MPI. In *Proc. ACM99*, 1999, electronic proceedings only, available at URL: http://www.crhc.uiuc.edu/~steve/wcbc99/.

MA Jie received his B.S. and M.S. degrees in computer science from Xi'an Jiaotong University in 1995 and 1998, respectively. He is currently a Ph.D. candidate of the Institute of Computing Technology, The Chinese Academy of Sciences. His research interests include high performance computer architecture, cluster operating system, high performance communication protocol and parallel computing.

HE Jin received his B.S. degree in material science and engineering from Xi'an Jiaotong University in 1996 and M.S. degree in computer science from Huazhong University of Science and Technology in 1999. He is currently a Ph.D. candidate of the Institute of Computing Technology, The Chinese Academy of Sciences. His research interests include high performance computer architecture, distributed file system and storage server.

MENG Dan received his B.S., M.S. and Ph.D. degrees in computer science from Harbin Institute of Technology in 1988, 1991 and 1995, respectively. He is an associate professor of the Institute of Computing Technology, The Chinese Academy of Sciences. His current research interests include high performance computer architecture, cluster operating system, high performance communication protocol, distributed file system and storage server.

LI Guojie received his B.S. degree in physics from Peking University in 1968, M.S. degree in computer science from University of Science & Technology of China in 1981, and Ph.D. degree in EE from Purdue University, USA in 1985. He is a member of Chinese Academy of Engineering and director of Institute of Computing Technology, The Chinese Academy of Sciences. His current research interests include high performance computer architecture, and MPP systems for bio-information processing.