GRID GATEWAY: ENABLING MESSAGE-PASSING IN A DISTRIBUTED COMPUTING ENVIRONMENT

Cui Wei, Ma Jie
Institute of Computing Technology, Chinese Academy of Sciences
P.O. Box 2704, Beijing 100080, P.R. China
Graduate School of the Chinese Academy of Sciences, Beijing 100039
{cw, majie}@ncic.ac.cn

ABSTRACT
The paper presents the flexible mechanism of a low-level inter-cluster communication protocol based on Grid Gateway, which enables external communication by delivering messages between clusters. Messages are forwarded in each cluster by Grid gateway nodes, which can act as computing nodes simultaneously. The number of gateway nodes varies from external communication load. The protocol can be used to transfer messages in local cluster. The multi-protocol method allows a traditional single-image of MPI to be laid over distributed clusters. The performance of inter-cluster communication based on Grid gateway is very high in the high-speed inter-cluster network so that geographically distributed clusters can be integrated through grid middleware and grid-enabled MPI environment into a high-performance computational Grid.

KEY WORDS
Inter-cluster Communication, Grid Computing, Grid-enabled MPI, Distributed Supercomputing

1. Introduction

Computational Grid provides more computational power than present machines, which meets the needs of many fields in science and commerce. Clusters with lots of applications and deployed plentifully play an important role in the Grid environment. To combine cluster computing with grid computing efficiently is discussed widely.

In some environment such as a campus or a building, some clusters are integrated through Grid middleware to perform distributed supercomputing. This computational Grid has some specific characters compared with a global grid platform. For example, inter-cluster network in the domain may be a specific high-performance network, inter-cluster communication has more intensive data transfer, and system configuration has some restricted roles. In order to communicate efficiently between clusters, protocols of cluster interconnects are connected through Grid Gateway (GGW) to exchange messages with each other directly, called external communication. The functionality of cluster low-level protocol is extended by GGW.

GGW has some flexible mechanisms, such as multi-gateway mechanism, and the dynamic of gateway nodes. These features improve scalability of inter-cluster communication and cluster scale. Grid Gateway also introduces some design issues in the new environment as follows, which are described in the paper.

- **Communication Semantic:** Extension of communication path and data-buffering at Gateway nodes result in some semantic issues of protocol, such as how to notify sender the send success, what sender does when the receiving buffers of gateways are unavailable.

- **Flow control:** Gateway stores messages of all the nodes before sending successfully, but its buffer blocks are very limited. Therefore a flow control mechanism is designed to avoid buffer overflow.

- **Multi-gateway mechanism:** Multiple gateways are used to improve the performance of total inter-cluster communication through load balance of external communication. The number of gateway in each cluster depends on the need of applications and system implementation.

This paper is organized as follows. Section 2 presents the architecture of GGW. Section 3 describes some design issues. Then the design and implementation of GGW is described in section 4. Section 5 presents performance evaluation. In the final sections relative researches are presented and the paper is summarized briefly..

2. Architecture of Grid Gateway

As shown in figure 1, distributed clusters are connected through inter-cluster network and make up the physical infrastructure of a computational Grid. Data transferred between clusters should pass three paths, computing nodes to gateway node via cluster system area network

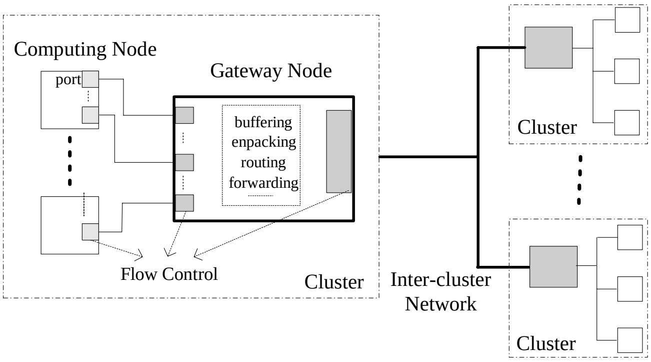

Fig. 1. Topology of Grid Gateway

(SAN), Gateway node to remote gateway node via inter-cluster network, remote gateway node to target computing node via SAN. On each gateway node, there is a Grid Gateway process delivering messages to the inter-cluster network or SAN.

Gateway processes and user processes use the same low-level protocol library to communicate each other as the library used by user processes to communicate with other user processes in a cluster. Computing nodes can be configured as gateway nodes manually for external communication. User processes use the intra-node communication of low-level protocol to transfer messages to gateway. Gateway process has blocking-receiving function so that it so doesn't affect the task running on the node when it becomes idle.

These may be more than one gateway nodes in a cluster responsible for delivering messages. At the extreme every computing node is a gateway node, which only transfers itself messages to remote clusters. Multiple gateways delivering messages parallelly can balance external communication loads. This multi-gateway mechanism allows a cluster to improve its efficiency of external communication when inter-cluster network is updated to support more hosts and bandwidth. Currently static strategy is used in the system and the configuration can be changed by hand.

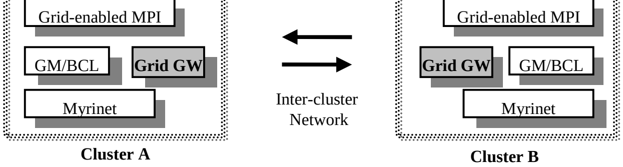

Fig.2. Structure of MPI Environment

GGW is developed to support MPI environment mainly in distributed high-performance clusters. User will have a single image of Grid-enabled MPI computing environment under the help of Grid middleware. Inter-cluster protocol has the same protocol interface with cluster protocol. It is relatively easy to provide an ADI over GGW for MPICH, a widely distributed implementation of MPI standard.

3. Design Issues

Grid Gateway enables separated low-level communication protocols to exchange messages between clusters directly. This mechanism extends the communication path and introduces complex buffer management and flow control mechanism because of high latency of whole communication path and the performance gap between cluster interconnects and inter-cluster network. Its flexibility for multi-gateway and gateway-dynamic mechanism needs more consideration in the design. These factors result in several design issues in the Grid Gateway.

3.1 Communication Semantic

A send operation in the internal communication protocol has a corresponding send-completed event after each sends when the message reach the receiver. Grid Gateway changes the semantic of send operation into an unreliable send operation as the result of the communication path extending to include local and remote cluster interconnects. If a send conforms to the semantic of the internal communication, high latency will result in many return events waiting for ACK/NACK. At last user has to be blocked temporally when the event queue reaches its capacity so that the node is unable to send and receive.

Fig.3 presents the data-moving paths of two kind of send operation GGW provided, control message transferring meta-information of primary message hided to users. Every GGW's send is followed by an event to tell user that the message gets to the gateway successfully or send fails, but user can not know if messages reach target computing nodes. This will impact the performance of the main user, MPICH. In the ADI over GGW, both rendezvous send and eager send have to provide the timeout/re-transportation mechanism. Receiver will return an ACK after receive a message. Fortunately, the time spent on waiting ACK can be hided by pipelining send operations.

The unreliable send relieves gateway processes from data management so that more CPU time can be saved. Buffer blocks can be free immediately after packets are sent out. Thus every gateway node can support more computing nodes to do external communication efficiently. In addition, this design allows gateway nodes to switch to other nodes dynamically in the future implementation.

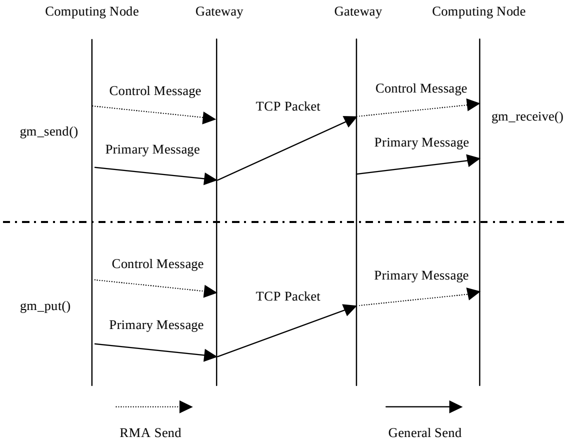

Fig. 3. Message Sequences in the Functions

3.2 Flow Control & Multi-gateway Mechanism

Because cluster SAN is high than inter-cluster network. There is performance gap between cluster interconnect and inter-cluster network. Moreover, gateway process may receive messages from many sources and forward to a few targets, resulting in buffer overflow. Thus, in case the buffer overflow, an efficient flow control mechanism is needed between computing nodes and gateway nodes, and between gateway nodes. Gateway uses TCP stream to communicate with remote gateway, so it is easy to design flow control, while flow control between computing nodes and gateway nodes is difficult relatively.

Grid Gateway use tokens to regulate sends between computing node and gateway node. A computing node port can send a message to gateway node only if when it possesses a token, and so does gateway. Tokens are returns after the messages arrive.

Each send token of computing node has a corresponding receiving buffer block on the gateway. The number of total send tokens must be least than the number of buffers. Since there are a lot of ports in the domain of a gateway, send tokens are assigned only to opened ports dynamically. When a new port is opened and initialized by gateway, some opened ports will hand over some tokens to it.

There are some parameters need to consider in the design, such as the number of tokens each opened port has at least, the number of buffers each gateway should provide, and the number of computing nodes in a gateway domain. User processes and gateway process use buffer ring to store control messages. The length of ring is equal to the number of send tokens of the port. Ring operation is a producer-consumer model. Receiver (or consumer) will update the ring head after receiving some messages and send a message to notice sender (or producer). In order to avoid sender being blocked to wait for free ring items, the ring length L should satisfy

$$\text{Maximum token number} \geqslant L$$
$$L \geqslant lat / g$$

where lat is the latency to transfer a control message, and g is the time between two successive sends. The two parameters can acquire from experiment.

Conventional job schedule strategy allows every processor executes a process, we assume every gateway can support P process to do external communication, and a process use only one gateway-initialized port. Therefore, the number of buffer N each gateway provided should satisfy following

$$N \geqslant P \times L$$

The value of P varies from implementations of practical systems and the need of applications for external communication performance. The system parameter can be configured manually. If a cluster has C processors, it will have C/P gateway nodes.

4. Design and Implementation

Grid Gateway is implemented based on the DAWNING 4000A high-performance Grid-enabling computer, which consists of 2560 Opteron processors and 640-port Myrinet. Currently we choose GM2.1.0 as low-level communication protocol of GGW. MPICH1.2.5 is ported to GGW by porting the GM ADI over GGW.

4.1 Host Communication Library

Communication end-point is identified by global ID, other than node ID used in GM, because GM node ID is a local unique identifier of end-pointer, and is not suitable in multi-cluster environment. Global ID is derived from the MAC address of interface.

Grid Gateway extends GM protocol API and adds some macro and functions for initialization, identifier conversion, etc. Some legacy functions are implemented as wrappers of GM functions, including send/receive, directed-send, open/close, etc. All functions can be used to send and receive message both in and between clusters.

Accompanying every message sent to remote clusters, control messages are sent to gateway nodes through RDMA instead of general send/recv, shown in fig.3. RDMA is a single-side operation so that the access to the ring of control messages needs not locks.

4.2 Gateway Process

Gateway process consists of three models: main procedure, receive-in thread and send-out thread. Main procedure collects information of local cluster and the remote, initiates those two threads, and blocks itself to accept new clusters' connection requests. Receive-in thread, shown in the upper part of Fig.4, repeatedly receives data from TCP/IP, enqueue it into receiving FIFO, and send it to its user processes through GM. Send-out thread, Figure 4's lower part, follows the same steps with receive-in thread but in the contrary direction.

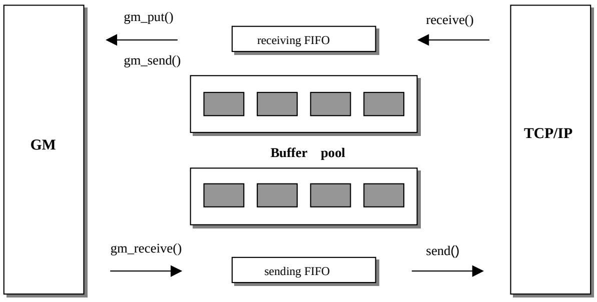

Fig. 4. Gateway Process Structure

Both two threads have own buffers so that gateway process is lock-free. Sending FIFO is a two-dimensional array, items of which are pre-allocated for every target gateway. Receiving FIFO is a two-dimensional list. A item is created only when receive-in thread gets a message because these are so many target nodes and ports that pre-allocation will waste memory.

5. Performance Evaluation

The performance of inter-cluster low level protocol based on Grid gateway depends on the location of user process and gateway process. When both processes are on one node and gateway delivers messages of local node, this delivering model is called Near Deliver (ND), otherwise is called Far Deliver (FD). It is possible that sender and receiver have different delivering model. For example, sender is Near Deliver (NSD) while receiver uses Far Deliver (FRD). The performance of MPI over Grid gateway, tested on the same platform using Pallas MPI Benchmark (PMB) suite, does not vary from delivering model obviously.

The testing platform consists of 8 nodes installed 64-bit Linux 2.4.19. Each node is a 2-way 1.6 GHz Opteron SMP with 133MHz PCI-X Bus. Myrinet uses M3F-PCIXD NIC and M3S-SW16-8F switch. GGW and MPI are tested on two inter-cluster networks, Myrinet with GM-IP (Myri) and Gigabit Ethernet (GbE).

5.1 GGW Performance

Peak bandwidth of GGW is 108Mbytes/s on GbE at 16KB message length, 199Mbytes/s on myrinet at 128KB, while the peak TCP bandwidths of inter-cluster network tested by Netperf are 118MB/s and 235MB/s respectively. Because intra-node bandwidth of GM is better than inter-node bandwidth, the bandwidth of NSD-NRD model for messages about from 1KB to 64KB is best. But for larger messages, bus competition between user process and gateway process in ND model makes FSD-FRD bandwidth becomes better than bandwidth of other models.

When multiple user processes perform external communication at the same time, the total bandwidth of all processes is equal to the peak bandwidth of single process.

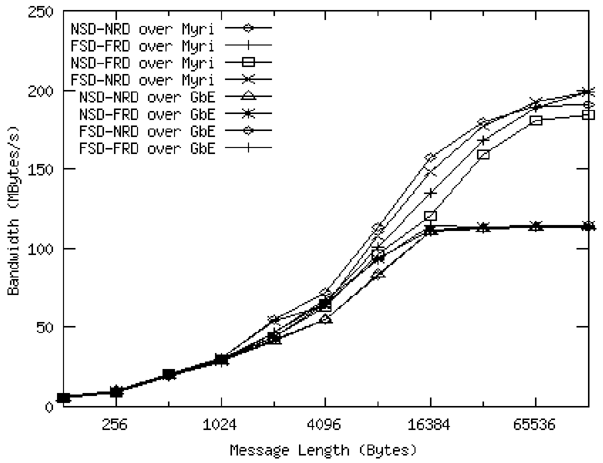

Fig.5. Ping-Ping Bandwidth of GGW

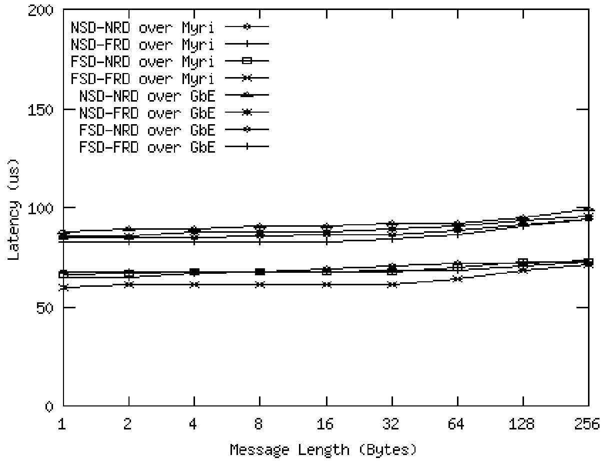

Fig.6. Ping-Pong Latency of GGW

Since TCP latency of GM-IP via Myrinet is about 20us better than Gigabit Ethernet, the latency of GGW over Myri is better than over GbE. The minimum is 60us and 81us respectively.

Delivering model also impacts point-to-point latency of GGW. The latency of NSD-NRD is 5-7us lower than FSD-FRD over both Myri and GbE.

As fig.7 shown, latency has some unavoidable overhead. Send overhead is the time from user process to gateway process, and vice versa is recv overhead. TCP overhead is the time spent on moving between gateways.

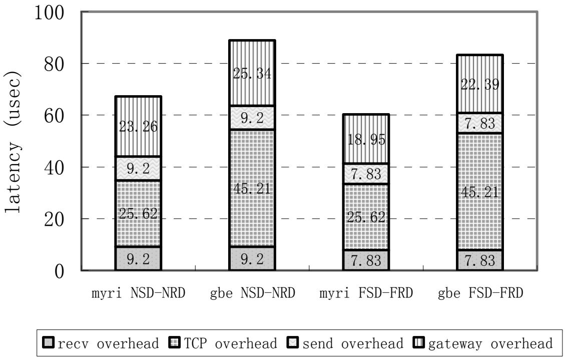

Fig.7. Latency Overhead of GGW

Most of time is spent on TCP and gateway overhead. Gateway process mainly operates on FIFO efficiently, but flow control has to limit user processes to issue messages.

5.2 Performance of MPICH over GGW

MPICH over GGW is only tested under NSD-NRD model on both inter-cluster networks, because the performance of all models is very close.

In PMB ping-pong testing between process p1 and p2, p1 send X bytes to p2, then p2 returns a message of same length. In ping-ping testing, p1 and p2 send X bytes to each other simultaneously. Latency is the time sending X bytes from p1 to p2 while bandwidth is X / latency.

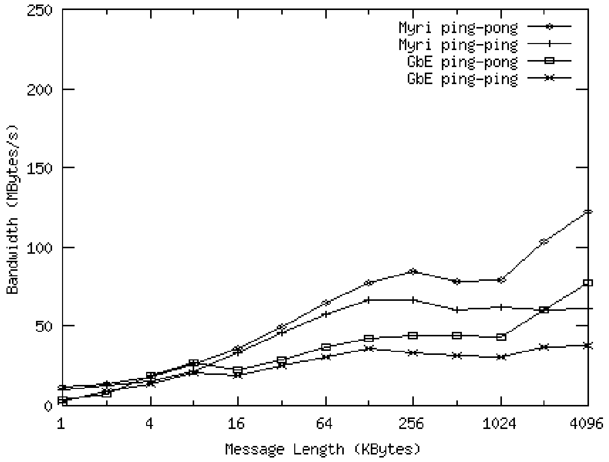

Fig.8. Pt2Pt Bandwidth of MPICH over GGW

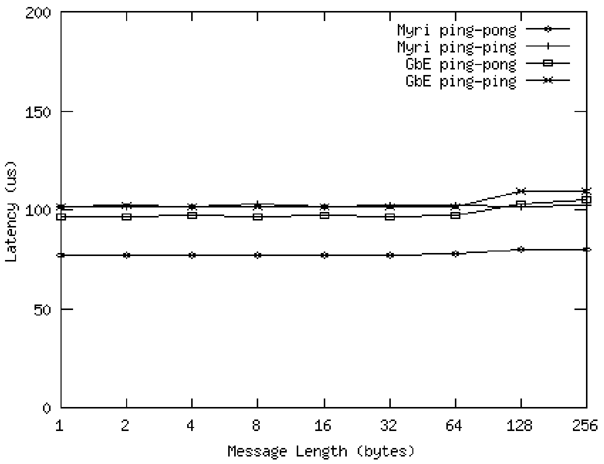

Fig.9. Pt2Pt Latency of MPICH over GGW

With Myrinet as inter-cluster network, peak ping-pong and ping-ping bandwidth of MPICH over GGW ADI are 122Mbytes/s and 60Mbytes/s respectively. On GbE, the two bandwidths are 77Mbytes/s and 38MB/s respectively. There is a decline from 256KB on because large message are divided into several segments by TCP.

Peak ping-pong and ping-ping latency are 76us and 101 us over myrinet, while 96us and 102us over GbE.

We compare GGW ADI with P4 ADI on Gigabit Ethernet in Tab.1. P4 reaches maximum bandwidth at 16KB length while GGW reaches it at 4MB. GGW is better than P4 for large messages, because transfer between gateways is overlapped with transfer between computing node and gateway so that host can send out large messages quickly.

Tab.1. PMB Pt2Pt performance comparison of MPICH over GGW ADI and over p4 ADI on Gigabit Ethernet

Performance	Packet Len	PingPing		PingPong	
		GGW	P4	GGW	P4
Latency(us)	1Byte	96.6	58.6	101.2	82.0
Bandwidth (MB/s)	16KB	22.3	73.3	19.5	55.5
	4MB	77.6	59.7	38.7	26.8

6. Relative Research

Many researchers have proposed and investigated communication mechanisms for distributed high-performance computers. Inter-cluster communication protocols mainly are provided in two ways: in the layer of low level cluster communication protocol or in MPI level.

Nexus runtime system provides multimethod communication on low level protocol directly. It can be extended to support many networks including Myrinet, ATM and Ethernet and so on, thus processes can use it to perform both external and internal communication. Integrating Nexus with the MPICH library can support large scientific application running in the heterogeneous networked environment.

MPICH-G2 is a widely distributed Grid-enabled MPI implementation. An ADI for MPICH based on Globus toolkits is implemented. MPICH-G2 also uses the vendor MPI libraries for internal communications. Two processes on different hosts have a socket connection for external communication.

PACX-MPI couples distributed high performance computer in a grid to provide a single MPI computing environment. Internal communications are mapped to vender MPI operations directly while external communications use a pair of daemon in each host to exchange data through TCP connection with that of target host.

7. Summary and Future Work

Based on Grid Gateway, distributed clusters are connected through a low-level inter-cluster communication protocol, over which MPI environment are provided. This paper describes its architecture and implementation. Several design issues are presented, including communication semantic, flow control and multi-gateway mechanism.

GGW can be applied flexibly, and allows the number and scale of involved clusters to increase dynamically while user processes communicate between clusters with high performance. GGW extends the functionality of low-level communication protocol so that it might facility the implementation of Grid-enabled tools. In the future plan, we want to apply GGW in large-scale environment. Grid middleware, Globus Toolkits or others, will be integrated with MPI over GGW to build a Computational Grid in specific environment with high performance.

REFERENCES

[1] I. Foster, C. Kesselman, S. Tuecke, The Anatomy of the Gird: Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications*, 15(3): 2001, 200-222.

[2] R. A Bhoedjang, User-level Network Interface Protocols, *IEEE Computer*, vol.11: 1998, 53-60.

[3] N. Karonis, B. Toonen, I. Foster, MPICH-G2: A Grid-enabled Implementation of the Message Passing Interface, *Journal of Parallel and Distributed Computing*, 63(5): 2003, 551-563.

[4] I. Foster, J. Geisler, etc, Managing Multiple Communication Methods for High-performance Metacomputing Applications, *Journal of Parallel and Distributed Computing*, vol.40: 1997, 35-48.

[5] E. Gabriel, M. Resch, T. Beisel, R. Keller, Distributed computing in a Heterogeneous Computing Environment, LNCS Vol.1497, 1998, 180-187.

[6] T. Imamura, Y. Tsujita, H. Koide, H. Takemiya, An Architecture of Stampi: MPI Library on a Cluster of Parallel Computers, LNCS Vol.1908, 2000, 200-207.

[7] Wei Cui, Jie Ma, Zhigang Huo, Grid Gateway: Message-Passing between Separated Cluster Interconnects, LNCS Vol.3032, 2003, 724-731.

[8] M. Muller, M. Hess, E. Gabriel, Grid enabled MPI solutions for Clusters, *The 3rd IEEE/ACM Int'l Symp. on Cluster Computing and the Grid (CCGRID2003)*, Tokyo, Japan, 2003

[9] Zhiwei Xu, Wei Li, Research on VEGA Grid Architecture, *Journal of Computer Research and Development*, 39(8): 2002, 923-929.

[10] V. G. Cerf, R .E. Kahn, A Protocol for Packet Network Intercommunicaiton, *IEEE Trans on Comms*, Vol. Com-22(5), 1974