An NAT-Based Communication Relay Scheme for Private-IP-Enabled MPI over Grid Environments

Siyoul Choi[1], Kumrye Park[1], Saeyoung Han[1], Sungyong Park[1], Ohyoung Kwon[2], Yoonhee Kim[3], and Hyoungwoo Park[4]

[1] Dept. of Computer Science, Sogang University, Seoul, Korea
{adore, namul, syhan, parksy}@sogang.ac.kr
[2] Korea University of Technology and Education, Chonan, Korea
[3] Sookmyung Women's University, Seoul, Korea
[4] Korea Institute of Science and Technology Information, Daejeon, Korea

Abstract. In this paper we propose a communication relay scheme combining the NAT and a user-level proxy to support private IP clusters in Grid environments. Compared with the user-level two-proxy scheme used in PACX-MPI and Firewall-enabled MPICH-G, the proposed scheme shows performance improvement in terms of latency and bandwidth between the nodes located in two private IP clusters. Since the proposed scheme is portable and provides high performance, it can be easily applied to any private IP enabled solutions including the private IP enabled MPICH solution for Globus toolkit.

1 Introduction

As cluster systems become more widely available, it becomes feasible to run parallel applications across multiple private clusters at different geographic locations as a Grid environment. However, in the MPICH-G2 library [1], an implementation of the Message Passing Interface standard over Grid environment, it is impossible for any two nodes located in different private clusters to communicate with each other directly across the public network until additional functions are added to the library.

In PACX-MPI [2], another implementation of MPI aiming to support the coupling of high performance computing systems distributed in a Grid, the communications among multiple private IP clusters are handled by two user-level daemons that allow the library to bundle communications and avoid having thousands of open connections between systems. However, since these daemons are implemented as proxies running in user space, the total bandwidth is only about half of the bandwidth obtained from kernel-level solutions [3]. It also suffers from higher latency due to the additional overhead of TCP/IP stack traversal and switching between kernel and user mode.

This paper proposes an NAT-based communication relay scheme, combining the NAT service with a user level proxy, for private IP enabled MPI solution over Grid environments. In our approach, only incoming messages are handled by a user-level proxy to relay them into proper nodes inside the cluster, while the outgoing messages are handled by the NAT service at the front-end node of the cluster. This brings

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 499–502, 2004.

performance improvement since we use the user-level proxy only once. By using the NAT service, which is generally provided by traditional operating systems, we can also easily apply our proposed scheme to any private IP enabled solutions without modifying operating system kernel. We have benchmarked our scheme and compared it with the user-level two-proxy scheme used in PACX-MPI [2] and Firewall-enabled MPICH-G [4]. The experimental results show that our NAT-based scheme outperforms the user-level two-proxy scheme.

The rest of the paper is organized as follows. Section 2 explains three communication relay schemes used for private IP enabled MPI, and provides the detailed mechanism of the NAT-proxy relay scheme. The experimental results are presented in section 3. Section 4 concludes the paper.

2 Communication Relay Schemes

In order to support the communication between private IP clusters in a Grid environment, we consider three communication relay schemes such as kernel-level two-proxy scheme, user-level two-proxy scheme, and NAT-proxy scheme.

In the kernel-level two-proxy scheme, we can implement a kernel-level proxy process in each of the front-end node within the cluster. Although this scheme is expected to have the best performance among the others described here, it is not used in general due to its poor portability.

In the user-level two-proxy scheme, we can implement a user-level proxy process in each of the front-end node within the cluster. A user-level proxy is easy to implement but has performance overheads such as those incurred by TCP/IP stack traversal and context switching between kernel and user mode. In this scheme, all the packets sent from one node to the other nodes located in other cluster have to go through the user-level proxy twice, which decreases the performance further. Despite its poor performance, this scheme has been widely used due to its highly portable nature. The PACX-MPI [2] and Firewall-enabled MPICH-G [4] use this scheme.

The NAT-proxy scheme is a combination of previous two solutions. The proxy implemented as a user-level program is responsible for forwarding only the incoming streams into the appropriate nodes within the cluster, while the outgoing streams go through the NAT service. Using a user-level proxy, no kernel modification is necessary. Moreover, since only incoming packets go through the proxy, the performance problems introduced by proxy can be minimized. Furthermore, using the NAT service for outgoing streams, multiple connections can be efficiently managed between front-end nodes of the clusters, which improves the communication performance further.

Fig. 1 depicts the NAT-proxy communication relay scheme proposed in this paper. In order to implement this scheme, each cluster should activate the NAT service in the front-end node. A user-level proxy, called stream relay daemon (SRD), is implemented in each front-end node. The SRD forwards incoming streams from the nodes in other clusters into their computation nodes. The outgoing streams from the computation nodes of one cluster go through the NAT service in the front-end node to reach the destination.

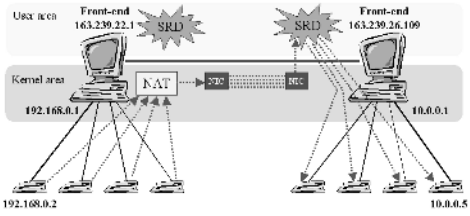

Fig. 1. The NAT-proxy communication relay scheme

3 Experimental Results

We have conducted our experiments over two private IP clusters, each of which has four computation nodes and one front-end node, respectively. The two clusters and all the nodes within the clusters are connected via 100Mbps Fast Ethernet cards. The two front-end nodes are configured to have both public and private IP addresses and each computation node is configured to have only private IP address.

In this benchmark, we compare the performance of our NAT-proxy scheme with that of the user-level two-proxy scheme. For the comparison, we measure the latency and the bandwidth between two private IP clusters using various traffic patterns.

Fig. 2 shows the latency between two private IP clusters. The latency was measured via *ping-pong* program using small sized messages (i.e., 128 bytes). As we can see from Fig. 2, our NAT-proxy scheme shows large performance improvement over two-proxy scheme by about 144%. For example, the measured latency using NAT and proxy was 1923 *usec*, while the latency using two user-level proxies was 2756 *usec*. It is clear from the result that the overhead incurred by using NAT was much lower than that of using two user-level proxies.

Fig. 3 compares the performance of our scheme with that of user-level two-proxy scheme by varying traffic patterns (one to one (1:1), many to one (2:1 and 4:1), and many to many (4:4) patterns) and varying message size from 1 Kbytes to 1024 Kbytes. As we can see from Fig. 3, the overall bandwidth obtained by using our scheme was much larger than that of using two user-level proxies. Furthermore, as we increase the message size, the performance gap is widening.

This can be explained by the following observations. In the user-level two-proxy scheme, the context-switching overhead (including message copy overhead between user space and kernel space) is bigger than that of our scheme, and the overhead becomes bigger as we increase the message size. If we apply the proposed relaying scheme to wide-area clusters, the performance improvement can be amortized to some extent, especially in small sized messages, due to the long delay (propagation delay) incurred between two front-end nodes. However, for the clusters transferring large messages and located in relatively near distance can benefit from the proposed scheme.

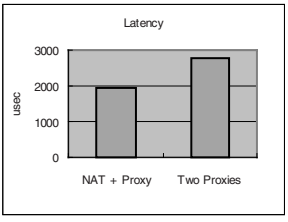

Fig. 2. Latency between two clusters

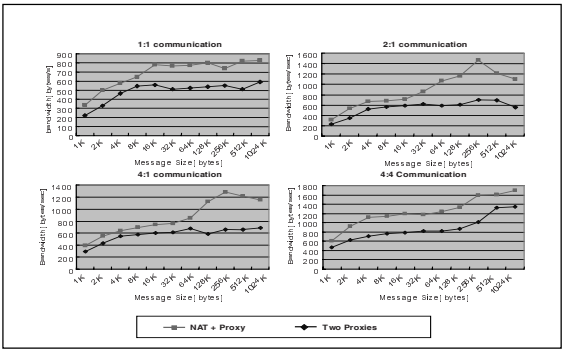

Fig. 3. Bandwidth between two clusters

4 Conclusion

In this paper, we have proposed a communication relay scheme based on the NAT and a user-level proxy, and compared our scheme with that of user-level two-proxy scheme that is implemented in PACX-MPI and Firewall-enabled MPICH-G. From the experiments, we showed that the performance of our scheme was better than that of user-level two-proxy scheme and the performance improvement became larger as we increase the message size. Considering that our scheme provides better performance and also does not require modifying kernel code to improve the performance, we can easily incorporate our scheme into any private IP enabled solutions.

Currently, we are working on developing a private IP enabled MPICH solution for Globus toolkit (i.e., MPICH-G2) using the scheme proposed in this paper.

References

1. Karonis, N.T., Toonen, B., Foster, I.: MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface (2002), `http://www3.niu.edu/mpi/`
2. Gabriel, E., Resch, M, Beisel, T., Keller, R.: Distributed computing in a heterogeneous computing environment, in Alexandor V., Dongarra, J. (eds.): Recent advances in Parallel Virtual Machine and Message Passing Interface, Vol. 1497 of Lecture notes of Computer Science, 180-188. Springer (1998). 5th European PVN/MPI User's Group Meeting.
3. Müller, M., Hess, M., Gabriel, E.: Grid enabled MPI solutions for Clusters, in Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'03) (2003)
4. Tanaka, Y., Sata, M., Hirano, M., Nakata, H., Sekiguchi, S.: Performance Evaluation of a Firewall-compliant Globus-based Wide-area Cluster System, in Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing, 121-128. IEEE Computing Society (2000)