論文の内容の要旨

論文題目　Design and Implementation of Scalable High-performance Communication Libraries for Wide-area Computing Environments
(広域計算環境における並列計算用のスケーラブルな高性能通信ライブラリの設計と実装)

氏名　斎　藤　　秀　雄

Over the past ten years, clusters have become the predominant architecture for performing parallel computation. By connecting multiple compute nodes by a Local Area Network (LAN), clusters make a large amount of processing power, memory and storage available for parallel computation. By connecting two or more of these clusters by a Wide Area Network (WAN), even more computational resources become available for parallel computation. Recently, the bandwidth of WANs has increased significantly, increasing the number of applications that can potentially take advantage of multi-cluster environments.

Unfortunately, multi-cluster environments are significantly more complex than single cluster environments. In particular, they introduce or magnify problems concerning connectivity, scalability, locality and adaptivity. As it is undesirable and unrealistic for each individual application to handle these problems separately, demands for wide-area communication libraries that handle these problems have increased.

Concerning connectivity, wide-area communication libraries need to be aware that connections between clusters are commonly blocked by firewalls or NAT. A simplistic scheme that assumes that all processes can connect to each other will encounter problems when deployed in WANs. Only some connections will be allowed, and messages must be routed between every pair of processes using those connections.

As for scalability, wide-area communication libraries need to avoid simplistic schemes that establish a large number of connections. While all connections consume resources, wide-area connections especially consume a lot of resources, causing various resource allocation problems. In addition to resource allocation problems, using a large number of wide-area connections in an uncoordinated fashion can result in low communication performance due to congestion.

The two previous requirements basically say that a wide-area communication library will establish connections between a subset of all process pairs. Then in order to maintain high communication performance, the process pairs that do establish connections should be selected in a locality-aware manner. In general, connections between nearby processes should be favored over connections between faraway processes, and connections between processes that communicate frequently should be favored over those

that communicate infrequently.

Moreover, wide-area communication libraries should automatically satisfy the three previous requirements by adapting to environments and to applications. They should not rely on manual configuration, because it is tedious, it does not scale, and it is the cause of various errors.

Much previous research has focused on each of these requirements separately, but more work is necessary in order for wide-area communication libraries to meet all of these requirements. For example, research centered around message passing offers good locality but poor scalability and adaptivity and research centered around P2P overlay networks offers good scalability and adaptivity but poor locality.

This has motivated me to make two proposals concerning the design and implementation of scalable high-performance communication libraries for wide-area computing environments: a locality-aware connection management scheme and a locality-aware rank assignment scheme. Using the two proposed methods, I have implemented a wide-area MPI library called MC-MPI and a wide-area Sockets library called SSOCK.

My connection management scheme overcomes firewalls and NAT by constructing an overlay network, and achieves scalability by limiting the number of connections that each process establishes to $O(\log n)$ when the number of total processes is n. In order to achieve high performance with a limited number of connections, the connections that are established are selected in a locality-aware manner, based on latency and traffic information obtained from a short profiling run.

My rank assignment scheme for wide-area communication libraries. This scheme finds a low-overhead mapping between ranks (process IDs) and processes by formulating the rank assignment problem as a QAP. It uses latency and traffic information obtained from the profiling run as well as routing information obtained from the connection management scheme in order to adapt to environments and applications.

Using the proposed connection management and rank assignment schemes, I have implemented a wide-area MPI library called MC-MPI. I have evaluated its performance by running the NPB on 256 cores distributed equally across 4 real clusters. For the IS benchmark, MC-MPI performed up to 2.1 times better than when connections were established between all processes. For the other benchmarks, MC-MPI was able to limit the number of process pairs that established connections to just 10 percent without suffering a performance penalty. Moreover, MC-MPI was able to find rank assignments that performed up to 4.0 times better than locality-unaware assignments and up to 1.2 times better than host name-based assignments.

In order to support not just MPI applications but any parallel application, I have also used the proposed connection management scheme to implement a wide-area Sockets library called SSOCK. In one experiment, SSOCK was able to connect 1,262 processes with each other in a 13-cluster environment with firewalls and NAT, without any of the connectivity issues and resource allocation problems that were encountered when the Socket library was used. In another experiment in which 100 processes simultaneously tried to establish connections, SSOCK was able to establish connections between all pairs of processes in 1.2 seconds, while the Socket library suffered from a large number of packet losses and timed out after 189 seconds.