MPICH-GP : 그리드 상에서 사설 IP 지원을 위한 MPI 확장[1)]

박금례 윤현준° 박성용
서강대학교 컴퓨터학과
namul@sogang.ac.kr, hjyun@dcclab.sogang.ac.kr°, parksy@sogang.ac.kr

MPICH-GP : An MPI Extension to Supporting Private IP Clusters in Grid Environments

Kumrye Park, Hyunjun Yun°, Sungyoung Park
Dept. of Computer Science, Sogang University

요 약

그리드 네트워크에서 MPI를 사용하여 지리적으로 산재된 컴퓨팅 자원을 활용하고 복잡한 문제를 해결하기 위한 MPICH-G2는 사설 IP 클러스터를 지원하지 못한다는 단점을 가지고 있다. MPICH-G2가 가지는 이러한 문제점을 해결하는 방법으로서 사설 IP 클러스터를 지원하는 MPICH-GP를 NAT와 프락시를 병용하여 설계하고 구현하였다. 사설 IP 클러스터의 프론트 노드에 프락시를 두고, 이를 통해 내부 계산 노드로의 통신 링크를 중계한다. 따라서 사설 IP 클러스터와 공인 IP 클러스터가 혼재된 네트워크에서도 적정한 경로를 설정하고 성공적으로 MPI 작업을 수행할 수 있다. 본 논문에서는 MPICH-GP의 성능을 기존의 MPICH-G2와 비교하였다. 그리드 환경에서 MPICH-GP는 MPICH-G2의 80% 이상의 성능을 보이며, *rank* 관리기법을 적용한 경우는 95%이상의 성능을 나타낸다.

1. 서 론

기존의 분산 및 병렬 컴퓨팅 환경은 지역적으로나 조직적으로 분리되어 있는 슈퍼컴퓨터, 대용량 저장장치, 다양한 고성능 연구장비와 데이터를 공유하는 그리드 컴퓨팅(Grid Computing)[1] 환경으로 급속히 진보해 나가고 있다. 기능적으로 그리드는 다양한 분산된 이기종의 컴퓨팅 자원과 데이터 자원을 단일한 형태로 보고 접근할 수 있도록 하고, 광범위하게 분산된 응용 시스템의 구축, 관리 및 사용을 위하여 대표적으로 글로버스 툴킷(Globus Toolkit)[2]이 사용된다. 한편 MPI(Message Passing Interface)[3]는 이러한 그리드 자원을 이용하여 복잡한 문제들을 해결하기 위해 사용되며, 이를 그리드 환경에 적용한 것으로 MPICH-G[4]와 MPICH-G2[5] 등이 있다. 이중 MPICH-G2가 표준적으로 사용되고 있다.

MPICH-G2는 기존 클러스터 환경을 그대로 그리드 환경으로 옮겨놓았기 때문에 각 노드들은 모두 공인 IP를 가지며 외부에 노출된다. 이 경우 외부에서 계산노드로의 직접 접근이 가능해, 보안 침해의 여지가 있다. 게다가 모든 계산노드에 공인 IP를 할당하는 것이 IP가 부족한 IPv4 환경에서는 문제가 될 수 있다.

사설 IP 문제 해결을 위한 기존의 방법으로 PAXC-MPI[6]와 MPICH-G가 있지만, PACX-MPI의 프락시와 MPICH-G에서 사용된 Nexus Proxy[7]는 일반적인 유저 영역에 구현되어 있기 때문에 상당한 성능저하의 요인이 된다.

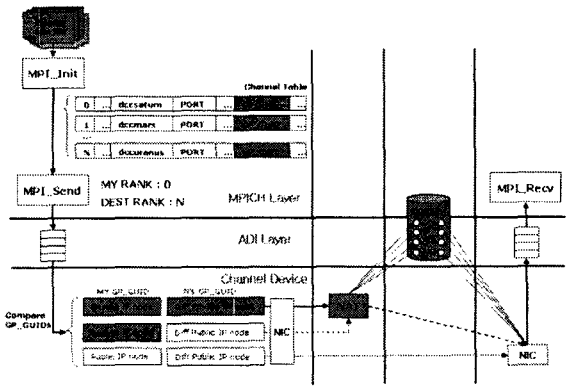
서로 다른 클러스터에 존재하는 사설 IP 노드간의 통신을 연결하기 위해서는 (사설 IP 계산노드)-(공인 IP 프론트노드)-(공인 IP 프론트노드)-(사설 IP 계산노드)를 거쳐야 한다. 이와 같은 상황에서 통신을 중계하기 위해 고려할 수 있는 방법들 중에서, 본 논문은 MPICH-G2를 수정하여 NAT(Network Address Translation)와 유저 영역에서 동작하는 프락시를 결합한 방법을 제안하고 이를 적용하여 MPICH-GP(Grid-enabled MPI supporting the Private IP cluster)를 구현하였다.

구현된 MPICH-GP의 성능은 클러스터의 구성방법을 제외하고는 동일한 조건으로 MPICH-G2와 비교되었고, 사설 IP 클러스터에서 통신을 중계하는 유저영역의 프락시 대몬이 가지는 성능 오버헤드와 응용 프로그램의 벤치마크를 통하여 실제 응용 프로그램에서 프락시에 걸리는 지연시간을 중점적으로 측정한다.

이후의 본 논문은 다음과 같이 구성된다. 2장에서는 사설 IP 클러스터의 통신을 중계하기 위하여 NAT와 프락시를 결합한 방법을 채택한 MPICH-GP의 중요 구성요소들에 대해서 설명한다. 3장에서는 LAN과 WAN 환경에서 MPICH-GP와 MPICH-G2의 성능을 벤치마크를 통해 알아본다. 마지막으로 4장에서는 논문의 결론과 향후 과제에 대해서 논하도록 한다.

2. MPICH-GP : 그리드 상에서 사설 IP 지원을 위한 MPI 확장

2.1 MPICH-GP 구조



[그림 1] MPICH-GP의 구조

MPICH-GP는 [그림 1]과 같은 구조를 갖는다. 초기화 단계와 실제 통신 단계로 나눠볼 수 있는데 프로세스간의 통신 단계에서는 경로 결정 단계와 경로에 따라서 실제 연결을 맺고 메시지를 전송하는 과정을 거치게 된다.

MPICH-GP의 초기화 단계에서는 파일을 통해 제공되는 참여 노드들의 정보를 통해 채널 테이블을 구성하게 된다. 이 과정에서 사설 IP 클러스터를 지원하기 위한 장치로서 사설 IP를 포함한 모든 노드들에 대해 고유한 ID인 GP_GUID를 생성하여 채널 테

---

1) 사사표시는 현재 논의중이며, 관리규정(협약 체결, 국고지원금 교부시 제시 예정)(안)에는

(국문 표기) "이 논문은 2006년도 두뇌한국21사업에 의하여 지원되었음.

(영문 표기) "This work was supported by the Brain Korea 21 Project in 2006." 로 되어있습니다.

이블 정보에 포함하게 된다. 이 정보를 통해 클러스터 내부에서만 유효한 사설 IP의 경우에도 각 노드를 구별하고 실제 통신할 수 있게 된다.

통신단계에서는 GP_GUID를 비교하여 연결 경로를 설정하는 과정과 이에 따라 연결을 맺고 실제 메시지를 전송하는 단계로 나눠볼 수 있다. 두 노드간의 통신 연결을 맺기 직전에 초기화 과정에서 구성한 채널 테이블의 GP_GUID를 비교하는 과정을 거친다. 이를 통해서 나와 상대 노드가 사설 IP를 가지는 노드인지 공인 IP를 가지는 노드인지를 비교하고 이에 따라 연결을 맺게 된다. 내가 사설 IP를 가진 노드인 경우는 프론트 노드에서 제공하는 NAT 서비스를 통해서 외부로 나아가게 되며, 상대가 사설 IP를 가진 노드인 경우에는 상대편 프론트 노드의 프록시를 통해 내부 노드로 들어간다.

2.2 사설 IP 클러스터의 통신 중계 방법

MPICH-GP에서는 외부로 나가는 연결은 NAT 서비스를 통해서 연결을 전달하며 내부 노드로 들어오는 연결에 대해서는 프론트 노드에 프락시를 통으로써 외부의 연결을 내부로 중계하는 방안을 채택하였다. NAT와 프락시를 병용한 중계 방안[8]은 커널 영역과 유저 영역의 중간적인 방법으로서 프락시를 통하여 외부로 나가는 연결과 내부로 들어오는 연결을 모두 처리하는 방안과 비교해서 성능이 우수하며, 커널 영역의 구현이 가지는 이식성의 문제를 해결할 수 있다.

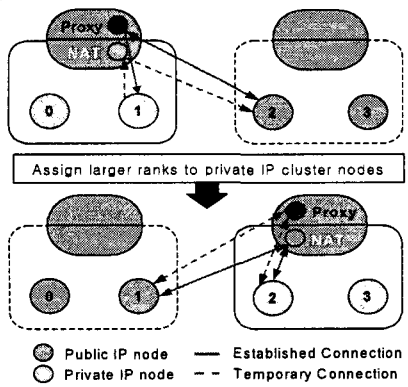
2.3 전역 ID 생성 관리

일반적으로 MPI 프로세스는 통신하려는 노드의 호스트명을 직접적으로 알지 못하고, 랭크(rank)를 통해 채널 테이블에서 정보를 참조하여 통신을 진행한다. 그러나 사설 IP 클러스터의 경우, 호스명만으로는 정확한 위치를 알아낼 수가 없기 때문에 프론트 노드의 호스트명, 계산 노드의 호스트명과 프로세스 ID로 구성된 GP_GUID를 통해 통신을 진행한다.

2.4 통신 프로토콜과 통신 중계를 위한 프락시

MPICH-GP는 기존 MPICH-G2에서 사용하는 프로토콜을 그대로 사용하여 호환성에 문제가 없도록 한다.

또한 MPI 프로세스는 집합 연산을 통해 다른 프로세스들과 거의 동시에 연속적으로 자료를 교환하는 특징이 있다. 프락시를 거치는 환경 하에서 이러한 연산이 효율적으로 동작할 수 있도록 글로버스가 제공하는 콜백 스페이스(callback space) 메커니즘을 이용한다.



[그림 2] Global Rank에 따른 연결 경로 변화

2.5 Rank 관리를 통한 성능향상 기법

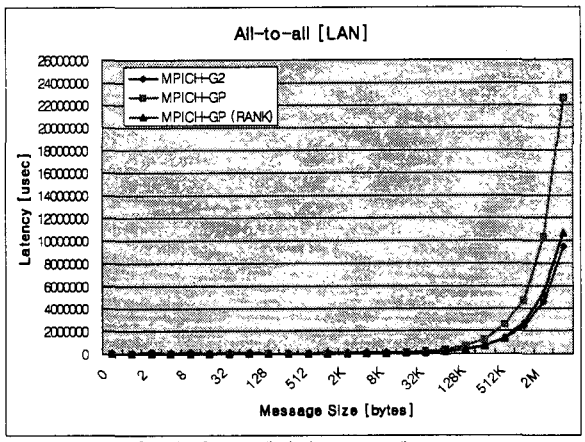
기존 MPICH-G2에서는 목적지 노드의 *global rank*가 소스 노드의 *global rank*보다 더 크면, 목적지 노드에게 다시 소스 노드로 연결을 맺도록 하고, 자신은 연결을 끊어버린다. 공인 IP 클러스터로 구성되는 MPICH-G2의 경우 이러한 방식은 성능에 거의 영향을 미치지 않는다. 그러나 프락시를 사용하며 사설 IP 클러스터와 공인 IP 클러스터를 모두 지원하는 MPICH-GP의 경우 *global rank*에 따른 연결 설정 메커니즘은 경우에 따라 성능에 큰 영향을 미칠 수 있다.

[그림 2]에서 보는 것처럼 기존 MPICH-G2에서의 연결 설정 메커니즘을 그대로 사용하게 되면, 사설 IP를 가지는 노드 1과 연결을 맺으려면 프락시를 통해야 하고 노드 1과 노드 2의 모든 메시지는 프락시를 통해 중계되어야 한다. 이것은 3장의 성능평가에서 살펴볼 수 있듯이 최초의 연결이 커널이 제공하는 NAT 서비스를 통해 중계되는 것과 비교하여 프록시를 거치는 오버헤드에 따른 성능 차이가 크다. [그림 2]의 아래쪽 그림은 이 문제에 대한 해결책을 제시하고 있다. 사설 IP 노드에 더 큰 *global rank*를 가지도록 조정하게 되면 그림에서 보는 것과 같이, NAT 서비스를 통해서 공인 IP를 가지고 있는 *global rank* 1인 노드로 직접 연결을 맺을 수 있다

3. 성능평가

본 장에서는 LAN, WAN환경에서 MPI의 통신 프리미티브와 응용 프로그램(각각 Pallas MPI Benchmarks(PMB)[9], NAS Parallel Benchmarks(NPB)[10])을 통해 MPICH-GP, MPICH-GP(RANK)의 성능을 분석한다.

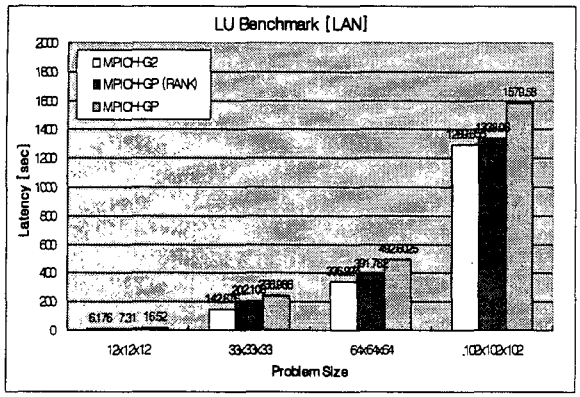
[그림 3]은 PMB에서 제공하는 다양한 벤치마크 중에서, 각각의 프로세스가 (각 프로세스에 대한 메시지 크기 × 프로세스의 수) 만큼 입력하고 받게 되는 가장 통신의 비중이 높은 Alltoall 테스트의 결과이다.



[그림 3] LAN에서의 Alltoall 테스트

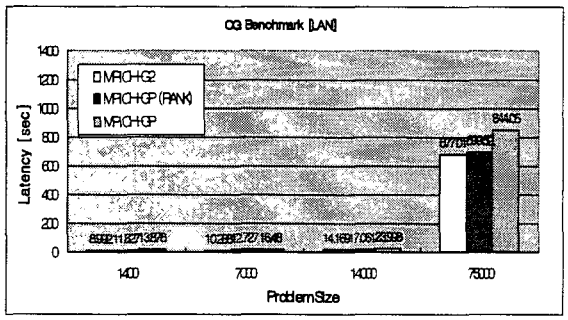
본 그래프에서는 메시지의 크기가 2배씩 증가함에 따라 지연시간은 지수적인 증가를 보이는데, 이것은 메시지 크기가 x일 때 한 프로세스가 다른 프로세스로 보내는 실제 메시지는 $4x$(메시지 사이즈 x × 프로세스의 개수)이기 때문이다. 그래프에서 NAT를 통해 메시지를 중계하는 *rank* 관리기법을 적용한 MPICH-GP의 경우는 MPICH-G2와 거의 비슷한 지연시간을 보이고 있지만, 프락시를 통해 메시지를 중계하는 MPICH-GP의 경우, 프락시로 과도한 메시지 중계에 따른 부하가 몰리기 때문에 이와 같이 지연시간이 가파르게 증가하는 그래프로 나타난다. 반면 WAN 환경은 전체적인 지연시간이 크게 증가하기 때문에 MPICH-GP(RANK)의 성능이 MPICH-G2와 동일한 성능을 나타낸다. MPICH-GP의 경우도 MPICH-G2의 75% 정도의 성능을 보인다.

[그림 4]는 NPB의 벤치마크 중 작은 메시지를 주고받으며 가까운 노드들과의 통신에 의존하는 LU의 벤치마크 이다.



[그림 4] LAN에서의 LU 벤치마크

이웃 노드들과의 통신이 많고 외부 클러스터 노드와의 통신이 적은 경우에 외부 클러스터 노드와의 통신을 중계하는 프락시의 부담을 측정할 수 있다. 문제의 크기가 커짐에 따라 MPICH-GP와 MPICH-G2의 지연시간 비율이 점차 줄어들며 102x102x102에서는 1.22로 MPICH-GP의 성능이 MPICH-G2의 80%정도로 나타난다. 이것은 MPICH-GP의 성능이 MPI 통신 프리미티브에서 측정한 결과보다 실제 응용 프로그램에서 좋은 성능을 낼 수 있음을 보여주는 것이다. MPICH-GP(RANK)는 MPICH-G2와 지연시간이 거의 동일한 비율로 증가하는 형태를 보이며 MPICH-G2의 95% 이상의 성능을 보인다. 마찬가지로 WAN 환경에서는 프락시의 통신 중계에 따른 지연의 성능 저하 영향력이 네트워크 지연 정도에 따라 둔화됨을 확인할 수 있다.



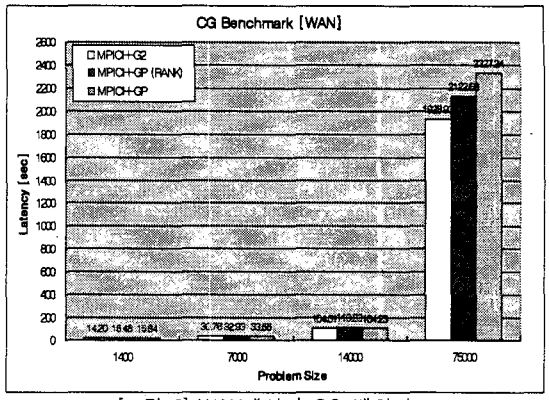
[그림 5] LAN에서의 CG 벤치마크

[그림 5]는 노드들 간에 통신량이 많은 CG 벤치마크의 결과이다. 작은 메시지의 경우에서 나타나는 지연시간 비의 큰 차이는 네트워크의 지연이 없는 LAN 상에서 프락시를 통과하는데 걸리는 시간이 전체 지연시간 중에서 큰 비중을 차지하기 때문에 나타나는 것이라고 말할 수 있다. MPICH-GP(RANK)는 NAT를 거치는 커널을 통한 메시지 중계에 용이하므로 지연시간 차이가 더 작게 나타남을 볼 수 있다.

[그림 6] WAN에서의 CG는 LAN에서의 평가와 비교하여 네트워크 속도 지연은 3배 이상 나타나며, 이것은 상대적으로 좋은 성능을 보여준다. 그러나 MPICH-GP의 경우 메시지 사이즈가 커질수록 프락시에서 연결을 중계하는 부하가 높아지게 되고 NAT의 도움을 받더라도 지연시간은 증가한다.

4. 결론 및 향후 과제

본 논문에서는 MPICH-G2가 가지는 문제점을 해결하는 방법으로서 사설 IP 클러스터를 지원하는 MPICH-GP를 NAT와 프락시를 병용하여 설계하고 구현하였다.



[그림 6] WAN에서의 CG 벤치마크

PMB[9]와 NPB[10]를 이용한 성능 평가에서 LAN에서의 성능 측정 결과 조금씩 차이는 있지만 RANK 관리기법을 적용한 MPICH-GP(RANK)의 경우는 MPICH-G2 성능의 90%, 적용하지 않은 MPICH-GP의 경우는 MPICH-G2의 50%정도의 성능을 보였다. 네트워크에서의 전파지연과 혼잡도가 LAN에서보다 훨씬 크기 때문에 통신 중계 장치인 프락시와 NAT에서 걸리는 지연시간은 어느 정도 가려지게 된다. 실제로 MPICH-GP(RANK)의 경우는 MPICH-G2 성능의 95% 이상의 성능을 보였으며, MPICH-GP의 경우는 평균적으로 MPICH-G2의 80% 가량의 성능을 나타냈다.

MPICH-GP는 그리드 환경에서 사설 IP 문제를 해결한다는 점에서 기여를 하지만, 일정 크기 이상의 메시지에 대해서는 압축을 해서 보내는 방식과 같은 개선이 필요하다. 또한 MPICH-GP의 성능을 개선하기 위해서는 프락시로 몰리는 부하를 분산시키기 위한 장치와 이에 대한 연구가 필요하다. 현재 많은 시스템들이 방화벽으로 묶여 있기 때문에 MPI를 위한 프락시에 방화벽을 넘어서 통신할 수 있도록 하는 것이 향후의 과제이다.

참고 문헌

[1] I. Foster, C. Kesselman and S. Tuecke. "The Anatomy of the grid : Enabling scalable virtual organizations". International Journals of Supercomputing Applications, 15(3), 2001

[2] I. Foster and C. Kesselman. "The Grid : A Blueprint for a New Computing Infrastructure". Morgan Kaufmann, 1998

[3] W. Gropp, E. Lusk, and A. Skjellum. "Using MPI: Portable Parallel Programming with the Message Passing Interface". MIT Press, 1995

[4] I. Foster, J. Geisler, W. Gropp, N. Karonis, E. Lusk, G. Thiruvathukal, and S. Tuecke. "A wide-area implementation of the Message Passing Interface." Parallel Computing, pp. 1735-1749, 1998

[5] N. Karonis, B. Toonen, I Foster, "MPICH-G2: a Grid-enabled implementation of the Message Passing Interface", Journal of Parallel and Distributed Computing, Volume 63, pp. 551-563, 1998

[6] Y. Tanaka, M. Sato, M. Hirano, H. Nakada, and S. Sekiguchi. "Performance evaluation of a firewall-complaint globus-based wide-area cluster system." In Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing, pp. 121-128. IEEE Computing Society, 2000

[7] I. foster and C. Kesselman, and S. Tuecke. "The Nexus approach to intergrating multithreading and communication. Journal of Parallel and Distributed Computing", pp. 70-82, 1996

[8] S. Choi, K. Park, S. Han, S. Park, "An NAT-Based Communication Relay Scheme for Private-IP-enabled MPI over Grid Environments", International Conference on Computational Science 2004 (ICCS 2004), pp 499-502, 2004

[9] Pallas MPI Benchmarks, http://www.pallas.com/e/products/pmb/

[10] NAS Parallel Benchmarks, http://www.nas.nasa.gov/Software/NPB