이질적인 NOW에서의 적응적 브로드캐스팅 방법

조 수 현° 김 영 학
금오공과대학교 대학원 컴퓨터공학과
{shcho, yhkim}@cespc1.kumoh.ac.kr

Adaptive Broadcasting Method on a Heterogeneous NOW

Soo-Hyun Cho° Young-Hak Kim
Dept. of Computer Engineering, Kumoh National University of Technology

요 약

슈퍼컴퓨터를 대신한 NOW는 고성능 계산작업의 대안으로 대두되고 있다. 일반적인 NOW 환경은 동일 성능의 노드들이기 보다 이질적인 노드들로 구성되어 있다. NOW 환경에서의 성능은 각 노드의 계산능력, 통신시간에 좌우될 수 있고 그밖에 전체 수행시간 관점에서 작업을 분할하는 방법 또한 중요한 요소 중 하나이다. 본 논문에서는 이질적인 성능을 가진 노드들로 구성된 NOW 환경을 고려한 적응적인 브로드캐스팅 방법을 제안하고, 그 밖에 성능 향상에 영향을 미치는 요소들을 성능평가를 통한 분석을 한다. 성능평가는 이질적인 노드들로 구성된 NOW 에서 LAM/MPI를 이용하여 측정하였다.

1. 서론

NOW(Network Of Workstation)는 네트워크에 연결된 PC, 워크스테이션 등의 계산능력을 공유하는 기술이기에, 참여하는 노드들의 성능은 이질적인 환경이 될 수 있다. 그러므로 NOW 환경에서의 병렬처리를 함에 있어 효율적인 집단화 통신이 무엇보다 중요하다.

일반적인 NOW 환경에서의 성능을 좌우하는 요소로는 다음과 같이 정리할 수 있다.

i. 노드들의 유휴시간을 줄임.
ii. 노드들의 부하를 효율적으로 분할.
iii. 전송 데이터의 크기 및 네트워크 상태 고려.
iv. 최적의 프로세스 수를 고려하는 것 등이 있을 수 있다.

위의 요소들에 대한 대부분의 연구들은 많이 진행되었지만 이질적인 노드들을 고려하지 않은 동일한 성능의 노드들에서의 집단화 통신에 관한 연구들에 불과하다.

따라서, 본 논문에서는 이질적인 노드들로 구성된 NOW 환경에서 노드들의 성능을 고려한 작업량을 배분하는 적응적인 브로드캐스팅 방법을 제안하며, 또한 작업 프로세스 수와의 관계를 성능분석 한다. 그리고 모든 성능평가를 위해서는 행렬 곱셈식과, LAM/MPI를 이용하였다.

본 논문의 구성은 다음과 같다. 2장은 관련연구에 대해 설명하고, 3장은 제안된 브로드캐스팅 방법에 대해 언급하며, 4장에서는 실험결과 및 분석에 대해서, 끝으로 5장에서 결론 및 향후 연구과제를 설명한다.

2. 관련 연구

일반적으로 이질적인 성능을 지닌 노드들과 네트워크 환경을 가진 병렬처리 시스템에서는 계산작업을 함에 있어 브로드캐스팅, 멀티케스팅 등과 같은 효율적인 집단화 통신 방법이 중요하다.

기존의 브로드캐스팅 방법에서는 근거리/원거리 기반의 효율적인 데이터 전송 방법에 대한 연구[1,2]들이 많이 진행되었다. 하지만 참여 노드들로의 빠른 데이터 전송관점이며, 동일한 성능에서의 같은 크기의 데이터 전송 방법들로 국한되어 있다.

또한 이질적인 노드들의 성능과 네트워크 환경을 고려한 연구[3,4]들에서는 효율적인 브로드캐스팅을 위한 스케줄링에 대한 연구들, 즉 데이터를 전송하기 전 가까운 노드들로의 재구성한다든지, 통신비용

을 고려한 송신자, 수신자를 선택한 후 동일한 크기의 데이터를 전송하는 방법들로 구성되어 있다.

본 논문에서는 참여하는 노드들의 성능을 파악한 후 각 노드별 전송 데이타의 크기를 차등적으로 송신하는 적응적인 브로드캐스팅 방법을 제안하여 전체 수행시간을 단축시키고자 하는 것이다.

3. 적응적 브로드캐스팅 방법

병렬처리를 함에 있어 노드들의 계산능력과 네트워크의 상태들을 고려하는 것이 중요하다. 또한 동일한 크기의 데이터를 전송하는 일반적인 브로드캐스팅 방법을 개선하여, 참여한 이질적인 노드들의 성능을 고려한 전송 데이터를 차등적으로 송신함으로서 성능향상을 꾀할 수 있을 것이다.

3.1 기존 방법

NOW 환경에서의 성능향상은 참여하는 노드들의 계산작업 외 각 노드들에게 보다 빨리 전송하는 것이 중요하다. Binomial Tree 구조를 가진 근거리 기반에서의 브로드캐스팅 방법이 일반적으로 다른 구조에 비해 가장 성능이 우수하다. 또한 원거리 기반의 브로드캐스팅을 함에 있어 효율적인 데이터의 전송 방법은 네트워크를 통한 불필요한 통신시간을 줄이고자 하는 것인데, 이 모든 것이 노드들의 성능은 고려하지 않은 동일한 환경과 노드들로 구성된 같은 크기의 데이터를 전송하는 방법들이다.

이질적인 네트워크 환경과 노드들의 성능을 고려함에 있어 데이터를 전송하기 전 FEF(Fastest Edge First), ECEF(Earliest Completing Edge First), look-ahead와 같은 알고리즘들이 제안되었다[4].

이 방법들은 참여하는 노드들에게 데이터를 전송하기 전 노드들의 성능에 따른 재구성을 한 후 동일한 크기의 데이터를 전송하는 방법이다.

다시 말해, 노드와 네트워크의 이질성을 고려한 통신 구조를 기반으로 한 것이지, 본 논문에서의 노드들 성능에 따른 다른 크기의 데이터 전송 방법은 고려하지 않고 있는 것이다.

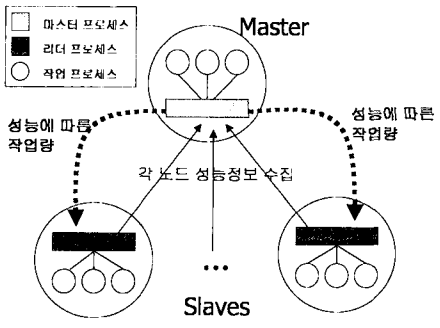

그림 1. 제안된 브로드캐스팅 방법

3.2 제안된 방법

그림 1과 같이 마스터 노드에서는 수행이전에 각 슬레이브 노드들의 성능정보를 파악한다. 본 논문에서는 일정 크기의 행렬식에 대한 노드들의 수행시간을 기준으로 각 노드들의 성능을 판단하게 되며, 노드들의 성능정보에 따라 마스터 노드는 성능에 따른 작업량을 슬레이브 노드들에게 전송하게 되는 것이다.

각 노드들의 성능에 따른 작업량 배분 정책은 수행능력에 대한 순위를 부여했을 때 상위 70%의 노드들에게 전체 작업량의 70%가 할당되고 나머지 노드들에게는 여분의 30% 작업량을 배분하는 것으로 하였다.

각 슬레이브 노드들에는 마스터로부터 작업을 받아 각 그룹의 작업 프로세스(Worker Process)들에게 전송하는 리더(Leader) 프로세스들로 구성 되어있다.

노드들의 성능을 고려하여 할당된 작업량이기에 리더 프로세스는 전송 받은 작업량을 그룹에 생성된 작업 프로세스 수만큼 균등하게 할당하게 된다. 그림 2는 본 논문에서 제안한 적응적 라이브러리의 구성을 나타내고 있다.

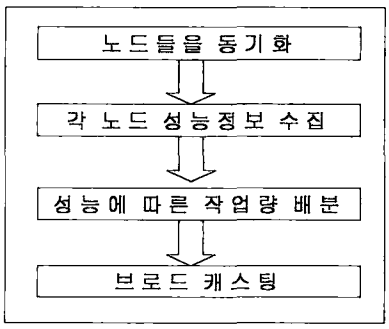

그림 2. 적응적인 브로드캐스팅 라이브러리 구성

4. 실험결과 및 분석

본 논문에서의 성능평가 방법은 제안된 브로드캐스팅 라이브러리를 동일한 네트워크 상태에서 전체 작업량과 프로세스 수와의 관계에 대한 평가 및 분석을 한다.

4.1 실험 환경

성능평가를 위한 실험환경은 다음과 같이 구성되어 있다.

운영체제	Linux 2.4.13
소프트웨어	LAM/MPI 6.3.2
CPU	PIII 733, PII 350, P150
Memory	24M, 32M, 64M, 128M

실험에 참여한 노드들은 10Mbps 이더넷으로 연결된 이질적인 성능을 가진 6대의 PC들로 구성되었으며, 성능평가를 위해 다양한 크기의 행렬 곱셈계산을 수

행하였다. 본 논문에서 성능평가 결과 값은 10번의 수행결과의 평균값을 이용하였다.

4.2 실험 결과

4.2.1 브로드캐스팅 결과 비교

그림 3은 노드 수와 작업 프로세스 수를 고정시킨 상태에서 전송되는 전체 작업량의 크기를 변화시켰을 때 기존방식[2]과 제안한 방식간의 결과 비교를 보여주고 있다.

작은 작업량(100, 200)을 수행 시켰을 경우 오히려 제안한 방식의 성능이 저조한 것을 알 수 있는데, 이것은 참여 노느들의 성능정보를 수집하는 절차가 추가되었기에 전체 수행시간 관점에서는 오히려 저조한 결과를 나타낸다. 하지만 순수 노드들에서의 작업 계산시간에서는 단축되었음을 알 수 있었다.

그밖에 작업량의 크기가 증가할수록 성능이 향상된다는 것을 알 수 있다. 이는 노드들의 성능정보 수집절차로 인해 추가된 시간을 작업량 증가에 따른 노드들의 성능을 고려하여 효율적으로 작업량을 배분한 결과이다. 즉, 순수 노드들의 작업 계산 시간이 많은 수행시간 단축을 가져왔기에 전체 수행 시간 면에서는 기존 방식보다 성능향상을 보이고 있는 것이다.

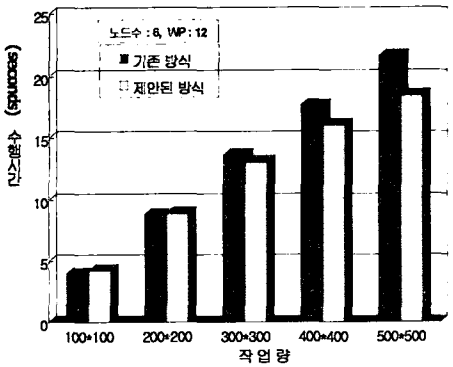

그림 3. 브로드캐스팅 결과 비교

4.2.2 작업자 프로세스 수에 따른 결과 비교

그림 4는 동일한 작업량과 노드 수에서의 작업 프로세스 수에 따른 실험 결과 비교를 보여준다. 실험 결과에서 최적의 작업 프로세스 수는 18개임을 알 수 있다.

하지만 18개 보다 많은 경우에 오히려 성능저하를 가져왔음을 결과에서 알 수 있다. 이것은 작업량을 처리하는 노드들의 처리시간과 마스터로의 결과 값 전송 시간은 일정한 것에 비해 작업 프로세스 수의 증가로 인한 통신시간이 상대적으로 소요되었기 때문이다. 그래서 각 노드들의 성능에 따른 최적의 작업 프로세스 수를 고려하여 불필요한 통신비용을 줄여야 할 것이다.

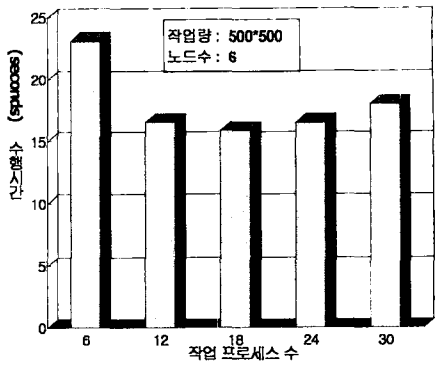

그림 4. 작업 프로세스 수에 따른 결과 비교

5. 결론 및 향후 연구과제

이질적인 노드들로 구성되어 있는 NOW 환경에서 각 노드들의 성능을 고려한 작업량을 배분하는 효율적인 간단화 통신 방법이 절실히 요구된다.

본 논문에서는 이질적인 노드들로 구성된 NOW 환경에서 노드들의 성능을 고려하여 작업량을 배분하는 적응적인 브로드캐스팅 방법을 제안하였다. 평가결과 전체 작업량이 증가할수록 성능이 향상되었다.

또한 작업 프로세스 수 관점에서는 작업 프로세스 수가 증가함에 따라 불필요한 통신시간으로 인한 성능저하를 가져왔다. 이는 노드들의 성능에 따른 작업량 배분 외 각 노드에 최적의 작업 프로세스 수를 생성시켜야 함을 알 수 있다.

향후 연구에서는 노드들의 성능에 따른 작업 프로세스 수와 결함 허용을 고려한 연구가 있을 예정이다.

참고 문헌

[1] T. Kielmann, R.F.H.Hofman, H.E.Bai, A.Plaat, and R.A.F.Bhoedjang, "MAGPIE: MPI's Collective Communication Operations for Clustered Wide Area Systems", In Proc. Symposium on Principles and Practice of Parallel Programming(PPoPP), pp.131-140, Atlanta, GA, 1999.

[2] 조수현, 김영학, "NOW 환경에서 작업자 프로세스의 수가 수행시간에 미치는 영향분석", 한국정보과학회 춘계학술발표대회 논문집(A) 제28권 제1호,pp.733~735,2001.

[3] M.Banikazemi, V.Moorthy, and D.K.Panda, "Efficient Collective communication on heterogeneous networks of workstations", In Proc. Intl.Conf. Parallel Processing, pp.460-467, 1998.

[4] Prashanth B. Bhat, C. S. Raghavendra, Viktor K. Prasanna, "Efficient Collective communication in distributed heterogeneous system", Proceedings of the 19th IEEE International Conference on Distributed Computing Systems, pp.15-24, May 1999.