

다중 메모리 모델의 CLUMP 시스템을 이용하기 위한 병렬 프로그래밍 기법과 성능 평가

이용욱^U 라마크리쉬나
광주과학기술원 정보통신공학과
{leeyu, chsung, cskim, jslee}@kjist.ac.kr

Parallel Programming for Exploiting Hybrid Parallel Model of CLUMP system and its Performance Evaluation

Yong-Uk LEE^U R. S. Ramakrishna
Dept. of Communication & Information
Kwang Ju Institute Of Science and Technology

요 약

클러스터를 구성하는 단위 노드로 SMP가 새로운 대안으로 시장에 등장하였다. 이러한 멀티프로세서 클러스터(CLUMP)는 하나의 시스템에 다중 메모리 구조를 가지는데, CLUMP가 가지는 다중 메모리 구조를 효과적으로 사용하기 위해서 본 논문에서는 중첩된 병렬화 프로그램 모델을 제안하였다. 중첩된 병렬화 모델은 중첩된 루프 레벨의 병렬화, 중첩된 태스크 레벨의 병렬화, 그리고 다중 중첩된 병렬화로 나뉜다. 본 논문에서는 중첩된 루프 레벨의 병렬화를 실험대상으로 하여 그 성능을 평가하고 단일 메모리 구조의 병렬화 프로그램과 성능을 비교하였다. 실험 결과 시험한 중첩된 병렬화 모델이 단일 메모리 구조의 병렬화 프로그램에 비하여 좋은 성능을 나타내었지만, 실험대상이 된 루프 레벨 병렬화의 잠재적인 특징으로 인해 실행에 참여하는 노드 수가 많아질수록 성능 향상 폭이 감소하는 결과를 보였다. 프로그램의 성능 향상 폭과 확장성은 문제 크기가 클수록 좋은 특성을 보였다.

1. 서론

현재 사용되고 있는 대부분의 클러스터 시스템은 확장성과 가격의 효율성 때문에 단일프로세서 노드를 네트워크로 연결한 분산 메모리 환경을 사용하고 있다. 그러나 시스템을 구성하는 프로세서, 메모리, 네트워크 인터페이스 기술의 발달로 2개나 4개의 프로세서를 가지는 소규모의 SMP 시스템의 등장하면서, SMP 시스템은 단일프로세서 노드를 대신하여 클러스터를 구축하는 새로운 대안으로 주목을 받기 시작했다. SMP 시스템을 이용한 클러스터 시스템을 CLUMP 라고 한다. CLUMP 시스템은 같은 개수의 프로세서를 사용하는 경우 단일프로세서 노드를 사용하는 클러스터에 비해 현저하게 네트워크 연결을 줄일 수 있고, 관리하기 쉽다는 장점을 가지고 있다.

CLUMP시스템은 다중의 메모리 구조를 가진다. 네트워크로 연결되어 있는 SMP 노드들은 분산 메모리 구조를 가지며 SMP 노드 자체는 공유메모리 구조를 가진다. 이렇게 다중의 메모리 구조를 가지는 CLUMP 시스템의 장점을 효율적으로 이용하기 위해서는 몇 가지 연구되어야 할 문제들이 있다. 가장 큰 문제는 CLUMP 시스템을 이용하기 위한 프로그래밍 모델의 제안이다. CLUMP 시스템의 다중 메모리 구조를 효율적으로 이용하기 위한

적당한 병렬화 프로그래밍 모델이 제안되어야 한다. 2장에서 우리는 현재 공유메모리와 분산메모리 환경의 프로그램을 위해 표준으로 사용되고 있는 프로그래밍 모델을 살펴보고, CLUMP 시스템을 효과적으로 프로그램하기 위하여 이들 프로그래밍 모델을 결합하는 방법을 제안하였다. 또 다른 문제는 다중 메모리 프로그래밍 모델이 기존의 단일 메모리 프로그램 모델과 비교해 어느 정도의 성능을 나타내는지 하는 것이다. 3장에서는 성능평가 실험을 위한 방법을 고찰하였고, 실험에 사용된 NAS Parallel Benchmark를 다중 메모리 프로그래밍 모델로 수정한 방법에 대해 언급했다. 4장에서는 실험 결과를 분석하고, 단일 메모리 모델과 비교함으로써, CLUMP 시스템의 종합적인 성능에 대해 고찰해 보았다.

2 CLUMP 프로그래밍 모델

2.1 단일 메모리 환경에서의 병렬처리 프로그램 모델

일반적으로 병렬 프로그래밍 모델은 프로세서와 메모리 모듈과의 연결에 의해서 결정된다. 전형적인 병렬처리 시스템은 분산 메모리 시스템과 공유 메모리 시스템이고 프로그램 모델로 각각 메시지 패싱 모델과 공유 메모리 모델을 사용한다. MPI는 이식성, 성능, 그리고 확장성이 좋아 현재 가장 널리 사용되는 메시지 패싱

프로그래밍 모델이다. 그러나 순차 프로그램과 다른 알고리즘이 사용되기 때문에 프로그램 개발이 어려운 것이 가장 큰 단점이다. 반면 OpenMP 프로그래밍 인터페이스는 공유 메모리 시스템의 병렬 프로그램을 위해 생긴 표준안이다.[4] OpenMP는 기존의 프로그램을 루프 레벨에서 병렬화 할 수 있기 때문에, 기존의 프로그램을 큰 수정 없이 바로 병렬화해서 사용할 수 있는 것이 가장 큰 장점이다. 또한 OpenMP는 공유 메모리 프로그램 모델 중 확장성과 이식성을 제공하는 유일한 프로그램 인터페이스이다. OpenMP는 컴파일러 디렉티브와 실시간 실행 라이브러리로 구성되어 있다.

2.2 중첩된 병렬화 프로그래밍 모델

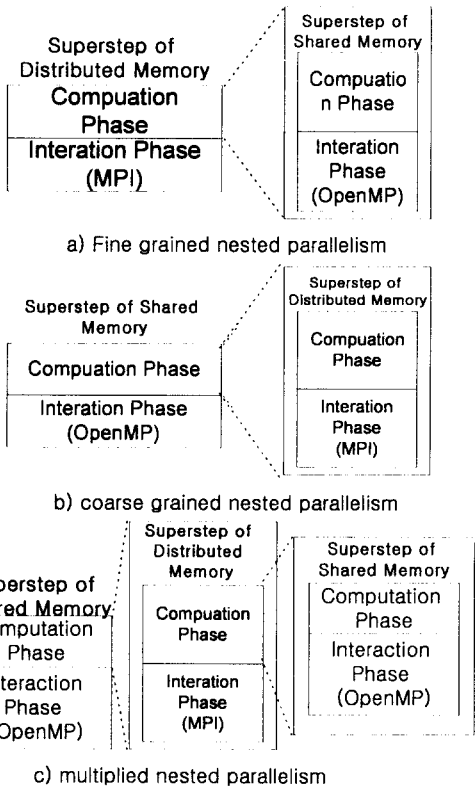


그림 1) 다중 메모리 모델의 중첩된 병렬화

효과적인 다중 메모리 구조의 프로그램을 위해 본 논문에서는 중첩된 병렬화 프로그래밍 모델을 제안하였다. L.G. Valiant의 병렬단계 모델에 따르면, 병렬연산은 순차적인 병렬처리 단계들로 이루어진다.[6][7] 각 병렬단계는 2개의 단계로 나뉘는데, 연산 단계와 상호작용 단계가 그것이다. 연산 단계에서 각 프로세서는 작업을 할당받아 맡은 작업을 수행하고, 상호 작용 단계에서는 각 프로세서간 통신과 동기를 수행한다.

다중 메모리 모델의 병렬 프로그래밍에서는 분산 메모리 환경과 공유 메모리 환경을 위한 두 개의 다른 병렬 단계가 존재한다. 이들 두 개의 병렬단계는 다중 메모리 구조의 시스템 내에서 세 가지 종류의 중첩된 병렬화 모델을 만든다. 첫 번째는 공유 메모리 모델의 병렬단계가 분산 메모리 병렬단계의 연산 단계에 중첩된 경우이다.(그림 1,a) 메시지 패싱에 의해 각 노드로 분산된 작업이 다시 각 노드내의 프로세서로 나뉘어져 중첩된 병렬화를 이룬다. SMP 노드 내에서 이루어지는 공유 메모리 병렬 단계에서는 주로 루프의 병렬화가 이루어진다. 이를 중첩된 루프 레벨의 병렬화라고 한다. 두 번째는 분산 메모리 모델의 슈퍼스텝이 공유 메모리 슈퍼스텝의 연산 단계에 중첩되는 경우이다.(그림 1,b) SMP 노드 내에서 생성된 쓰레드가 독립적인 메시지 패싱 프로그램을 수행하여 병렬화를 이룬다. 공유 메모리 병렬단계에서는 주로 작업의 병렬화가 이루어진다. 이를 중첩된 태스크 레벨 병렬화라고 한다.

마지막으로 위의 두 가지 종류의 슈퍼 스텝이 서로 중첩된 경우이다.(그림 1,c) SMP 노드 내에서 생성된 쓰레드가 독립적인 메시지 패싱 프로그램을 수행하고 메시지 패싱 내부는 다시 루프 레벨로 병렬화 되는 구조이다. 이를 다중 중첩된 병렬화라고 한다. 다중 중첩된 병렬화는 다른 두 개의 중첩된 병렬화의 약점을 상호 보완하기 위해 사용될 수 있다.

3. 실험 방법

3.1 실험 장비와 실험 대상

본 실험은 컴팩의 HPC320 시스템을 이용하여 수행하였다. HPC320은 4개의 CPU를 가진 ES40 SMP 시스템을 단위 노드로 하여 8개의 SMP 노드를 메모리 채널로 묶은 32개 CPU를 가진 SMP 클러스터이다.

또, 다중 메모리 프로그래밍 모델의 성능을 평가하기 위해 우리는 NASA의 Ames 연구 센터에서 개발한 NAS 병렬 벤치마크(NPB) 프로그램을 사용하였다. NPB는 현재 분산 메모리 병렬 시스템의 성능을 측정하고 비교하기 위해서 많이 사용되는 벤치마크 프로그램이다.

본 실험에서는 커널 프로그램 중 EP를 실험 대상으로 하여 다중 메모리 모델의 병렬화를 수행하였다. 앞에서 언급한 중첩된 루프 레벨 병렬화가 EP의 다중 병렬화를 위해 적용되었으며, 다중 병렬화 프로그램의 메시지 패싱 부분을 위해서는 MPI가 고유 메모리 부분을 위해서는 OpenMP가 사용되었다.

4. 중첩된 루프레벨 병렬화의 성능 평가

실험은 다섯 개의 문제 크기에 대하여 반복되어졌다. 매 실험마다 1개에서부터 8개까지의 노드를 사용할 때의 성능이 측정되었다. SMP노드 내에서 OpenMP에 의해 생성되는 쓰레드는 2개에서 4개까지 변화를 주며 실험했다. 반면에 단일 메모리 구조의 병렬 프로그램 성능은 수정되지 않은 NAS의 EP를 다섯 개의 문제 크기에 대하여 1개에서 8개까지 노드를 변경하며 실행하며 그 결과를 측

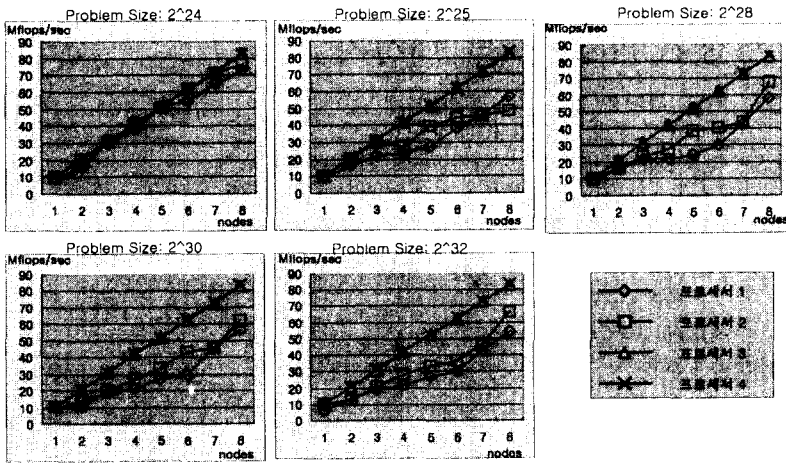


그림 2) 다중 메모리 모델의 성능 평가

정하여 구할 수 있었다.

실험 결과는 그림 2)와 같다. 종합적으로 다중 메모리 환경에서의 중첩된 병렬화 프로그래밍 모델은 단일 프로세서를 사용하는 단일 메모리 구조, 병렬화 프로그램에 비해 높은 성능을 나타내었다. 노드 당 쓰레드가 할당된 프로세서가 늘어날수록 성능도 비례하여 증가하였으며, 실행에 참여하는 노드 수가 증가할수록, 성능도 같이 상승하였다. 성능 향상은 3개 이상의 프로세서를 사용하는 경우에는 단일 프로세서를 사용하는 경우에 비해 문제 크기가 큰 경우에는 평균 1.7배, 문제 크기가 작은 경우에는 평균 1.5배의 성능 향상을 보여주었다. 2개의 프로세서를 사용하는 경우에는 이 보다 훨씬 적은 평균 1.2배 이하의 성능 향상을 보여주었다.

가장 작은 문제 크기를 가지는 경우에는 사용한 프로세서의 수에 상관없이 거의 모든 경우에 거의 동일한 성능 향상을 보이는데, 프로세서의 추가로 얻어지는 루프 크기 감소가 쓰레드의 생성 시간과 상쇄되어 얻어진 결과로 생각된다.

프로세서를 한 개 사용한 단일 메모리 병렬 프로그래밍 방법의 경우, 노드 수에 따라 지수적으로 성능이 증가하는 반면에, 프로세서를 3개나 4개 사용한 다중 메모리 병렬 프로그램에서는 노드 수에 따라 성능이 선형적으로 증가하였다. 이것은 노드 수가 증가할수록 여러 개의 프로세서가 나누어 가지는 작업 크기가 줄어들기 때문에 나타나는 성능향상 폭의 저하라고 생각된다.

5. 결론

지금까지 다중 메모리 구조에서의 사용되는 병렬 프로그래밍 모델을 제안하고, 제안된 병렬 프로그램 모델을 이용하여 HPC320 시스템의 성능을 평가하였다. 또한 평가한 내용을 단일 메모리 구조의 병렬 프로그램과 비교하여 그 결과를 분석하였다. 실험은 중첩된 루프 레벨의

병렬화를 이용하여 이루어졌다. 실험 결과 전체적인 프로그램의 성능은 실행에 참여하는 노드 수에 따라 변화가 심했다. 이는 중첩된 루프 레벨 병렬화의 잠재적인 취약점으로 고정 문제 크기인 경우 노드 수가 증가할수록 각 프로세서에 할당되는 루프크기가 줄어들기 때문에 이상적인 성능 향상을 기대할 수 없는 것이다. 따라서 실행에 참여하는 노드 수와 상관없는 시스템의 성능 향상을 꾀하려면 근본적으로 다중 중첩된 병렬화가 고려되어야 할 것이다.

7. References

[1] Cappello, F.; Richard, O.; Etienne, D. "Investigating the performance of two programming models for clusters of SMP PCs", High-Performance Computer Architecture, 2000. HPCA-6. Proceedings. Sixth International Symposium on , 1999 , Page(s): 349 -359

[2] Dagum, L.; Menon, R. "OpenMP: an industry standard API for shared-memory programming", IEEE Computational Science and Engineering, Volume: 5 1 , Jan.-March 1998 , Page(s): 46 -55

[3] Clark, D. "OpenMP: a parallel standard for the masses", IEEE Concurrency [see also IEEE Parallel & Distributed Technology], Volume: 6 1 , Jan.-March 1998 , Page(s): 10 -12

[4] Hu, Y.C.; Honghui Lu; Cox, A.L.; Zwaenepoel, W. "OpenMP for networks of SMPs", 13th International and 10th Symposium on Parallel and Distributed Processing, 1999, Page(s): 302 -310

[5] OpenMP Organization. "Fortran Language Specification, v. 1.0", <http://www.openmp.org/specs/> October 1997.

[6] Leslie G. Valiant "A bridging model for parallel computation", Commun. ACM 33, 8 (Aug. 1990), Pages 103 -111 [Find Related Articles]

[7] Yuzhong Sun; Jianyong Wang; Zhiwei Xu "Architectural implications of the NAS MG and FT parallel benchmarks", Advances in Parallel and Distributed Computing, 1997. Proceedings , 1997 , Page(s): 235 -240

[8] Cappello, F.; Richard, O.; Etienne, D. "Investigating the performance of two programming models for clusters of SMP PCs", High-Performance Computer Architecture, 2000. HPCA-6. Proceedings. Sixth International Symposium on , 1999 , Page(s): 349 -359

[9] Gropp, W.W.; Lusk, E.L. "A taxonomy of programming models for symmetric multiprocessors and SMP clusters", Programming Models for Massively Parallel Computers, 1995 , 1995 , Page(s): 2 -7