이기종 시스템으로 구성된 클러스터 시스템을 위한 MPI Work Packet Manager

*이규호, 김준성
중앙대학교 공과대학 전자전기공학부
John Morris
Department of Electrical and Electronic Engineering
The University of Auckland
e-mail : zodiac83@wm.cau.ac.kr, {junkim, morris}@cau.ac.kr

MPI Work Packet Manager: Exploiting Heterogeneity in Cluster Computing

*KyuHo Lee, JunSeong Kim
School of Electrical and Electronics Engineering,
College of Engineering, Chung-Ang University
John Morris
Department of Electrical and Electronic Engineering,
The University of Auckland

Abstract

저가의 개인용 컴퓨터를 고속 네트워크 장비와 시스템 독립적인 통신 라이브러리를 이용하여 연결함으로서 병렬처리 프로그램을 수행할 수 있는 클러스터 시스템을 구축할 수 있다. 클러스터 시스템은 이를 쉽게 구성하는 개별 시스템의 교체 및 추가로 인해서 이기종화 된다. 본 논문에서는 이기종 시스템으로 구성된 클러스터 시스템의 성능 향상을 위해서 MPI Work Packet Manager (WPM) 라이브러리를 제안한다. 실험결과에 의하면 이기종 시스템으로 구성된 클러스터 시스템에서 WPM 라이브러리를 적용한 병렬처리 프로그램이 이기종 클러스터 시스템의 성능을 적절히 활용하고 있음을 알 수 있다.

I. 서론

클러스터 컴퓨팅이란 분산 병렬 컴퓨팅으로서, 분산 컴퓨팅은 하나의 큰 문제를 네트워크로 연결된 여러 개의 프로세서를 이용하여 해결하는 방식이며, 병렬 컴퓨팅은 하나의 큰 문제를 여러 개의 태스크로 나누어 한꺼번에 수행함으로서 해결하는 방식이다. 기술의 발전으로 개인용 컴퓨터 또는 워크스테이션과 간단한 네트워크 장비들로 클러스터 시스템을 구축하기 용이해 졌다. 그러나 급속한 기술 발전은 더 좋은 성능의 프로세서를 지속적으로 생산할 수 있도록 하였으며, 이것은 프로세서의 교체 주기를 짧게 하여 클러스터 시스템의 이기종화를 가져왔다[5]. 이렇게 이기종화된 클러스터 시스템을 이용한 병렬처리 시스템의 경우 클러스터 시스템을 구성하는 개별 컴퓨터 또는 워크스테이션의 성능을 고려해야 그 성능을 최대한 이용할 수 있다.

몇몇 프로그래밍 언어와 라이브러리가 클러스터 컴퓨팅 환경에서 병렬처리를 수행할 수 있도록 개발되었다. 그 중에서 클러스터 시스템을 기반으로 하는 병렬처리 시스템 구축에 가장 많이 사용되고 있는 MPI의 경우 특별한 work scheduler를 포함하지 않고 기본적인 메시지 전송 메커니즘만을 제공하고 있다. 따라서 이기종 클러스터 시스템을 기반으로 하였을 때는 병렬처리 프로그램들이 개별 컴퓨터의 성능을 최대한 활용할 수 있도록 설계하고 구현해야만 한다. 본 논문에서는 Work Packet Manager (WPM)이라는, 이기종 클러스터 시스템을 구성하고 있는 개별 시스템들의 성능을 최대한 활용할 수 있고 병렬처리 프로그램 개발시 쉽게 사용할 수 있는 MPI 프로그램에 사용될 수 있는 work 스케줄러를 제안한다.

본 논문의 2장에서는 WPM의 비교대상인 Cilk와

WPM의 기반인 MPI에 대해서 간략하게 살펴보고, 3장에서는 WPM에 대한 자세한 설명을, 4장에서는 WPM을 적용한 프로그램의 성능 측정을 위한 실험 환경과, 실험 결과를 분석한다. 마지막으로 5장에서 결론을 내린다.

II. 배경

2.1 Cilk Multithreaded Runtime System

Cilk는 병렬처리를 위해 개발된 프로그래밍 언어와 그 언어를 이용하여 작성된 프로그램을 실행할 수 있는 환경을 제공하는 runtime 시스템이다[2,3]. Cilk는 일반적인 병렬처리 프로그램 개발을 위해 디자인 되었지만 dataflow 모델을 기반으로 하여 병렬적인 데이터 형태 또는 메시지 교환 형태로 작성하기 어려운 병렬처리 프로그램을 개발하는데 더욱 효과적이다. Cilk runtime 시스템에서 제공하는 dataflow 모델은 쓰레드를 생성하고, 생성된 쓰레드에서 필요한 데이터의 준비가 완료되면, 자동적으로 쓰레드와 데이터를 비어있는 프로세서에서 가져가 처리하도록 함으로서 병렬처리를 수행하는 형태이다. Cilk 언어는 기본 C 언어(ANSI C)에 예약어를 추가한 형태로 runtime 시스템에서 데이터 관리, 메시지 관리, 계산 및 동기화 쓰레드 관리 등을 처리할 수 있도록 구성되어 있다. 이렇게 함으로서 C 언어를 알고 있는 프로그래머는 쉽게 Cilk 프로그래밍 언어를 사용할 수 있고, 복잡한 부분에 대한 관심을 가질 필요 없이 쓰레드의 생성에만 관심을 기울이면 된다.

2.2 Message Passing Interface (MPI) Standard

MPI는 메시지 교환을 통해 병렬처리를 수행하는 프로그램을 개발하기 위한 라이브러리 표준이다.[1,2] 현재 대학, 연구소, 산업체 등을 포함한 40여개의 단체가 참여하고 있는 MPI Forum에서 개발되었고 유지되고 있으며 병렬처리 시스템 개발에 가장 많이 사용되고 있다. 본 연구에서는 MPI 표준을 구현한 라이브러리들 중에서 MPICH를 이용한다.[4] MPI 라이브러리를 이용해 개발된 병렬처리 프로그램은 send와 receive 명령어를 통해 각 컴퓨터 간에 메시지를 교환함으로서 병렬처리를 수행한다. 라이브러리에서 기본적으로 이용할 수 있는 함수는 MPI_Init, MPI_Comm_rank, MPI_Comm_size, MPI_Send, MPI_Recv, MPI_Finalize의 6개이며, 필요에 따라 추가적으로 125가지의 다양한 함수를 이용할 수 있다. MPI 라이브러리를 이용해 개발된 프로그램을 수행하기 위한 실행 환경은 기본적인 메시지 교환 환경만을 제공하며, 특별히 준비된 작업 관리 시스템은 없다. 따라서 프로그래머가 직접 작업 관리 시스템을 설계/구현할 필요가 있다.

III. Work Packet Manager

본 논문에서는 MPI를 이용한 병렬처리 프로그램의 성능을 향상시키고, 보다 편리하게 프로그램을 작성할 수 있도록 하기 위해 Work Packet Manager (WPM)을 설계하고 구현한다. 병렬처리 프로그램을 작성함에 있어서, 개별 시스템에 같은 양의 work를 할당하는 것은 이기종으로 구성된 클러스터 시스템에서는 자원을 낭비하는 결과를 초래한다. 따라서 WPM에서는 개별 컴퓨터의 성능을 고려한 work 분배를 할 수 있도록 구현하였다.

WPM은 WorkPacket, WorkQueue, WorkManager로 구성된다. WorkPacket은 하나의 work를 저장하는 저장소로서 개별 컴퓨터 사이의 데이터 교환에서 기본 단위가 된다. WorkQueue는 WorkPacket들을 저장하기 위한 공간으로서 큐의 형태를 가지고 있어 먼저 들어온 WorkPacket을 먼저 내보내는 FIFO 구조를 가진다. WorkManager는 WorkQueue들을 관리하기 위한 것으로서 내부에 처리해야할 WorkPacket을 저장하는 Source큐와 처리된 WorkPacket을 저장하는 Result큐를 가지고 있다. 표 1에 WorkPacket, WorkQueue, WorkManager의 구조를, 그림 1에 WPM을 이용한 병렬처리 프로그램의 개략적인 흐름을 나타내었다.

표 1. WorkPacket, WorkQueue, WorkManager 구조

WorkPacket

```
struct workpacket_t {
  unsigned size;
  void *data;
  struct workpacket_t *next;
};
```

WorkQueue

```
struct workqueue_t {
  sturct workpacket_t *head;
  unsigned wp_size;
  unsigned wq_length;
};
```

WorkManager

```
struct workmanager_t {
  struct workqueue_t *source
  struct workqueue_t *result;
  unsigned datasize;
};
```

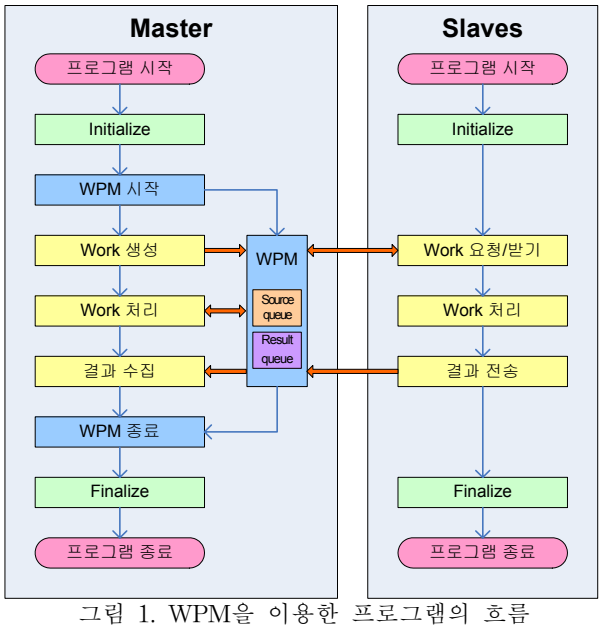

그림 1. WPM을 이용한 프로그램의 흐름

Ⅳ. 성능 측정

4.1 실험 환경

6대의 서로 다른 성능을 가지는 개인용 컴퓨터로 구성된 이기종 클러스터 시스템이 실험에 사용되었다. 각 컴퓨터들은 최소한 128Mbyte의 메모리를 가지도록 설정했으며, 페이징이 필요할 만큼 큰 프로그램을 실험에 사용하지 않으므로 디스크 성능은 고려하지 않는다. 각 컴퓨터는 100Mbps로 동작하는 Nortel BayStack 70-24T 이더넷 스위치로 연결된다. 표 2에 실험에 사용된 시스템 설정을 종합하여 보여준다.

표 2. 실험에 사용된 클러스터 시스템 구성

이름	동작클럭 (MHz)	메모리 (Mbytes)	운영체제 Linux
ant1	400 (P-II)	128	v7.3
ant2	400 (P-II)	128	v7.3
ant7	450 (P-III)	128	v7.3
ant9	800 (P-III)	128	v9.0
ant11	800 (P-III)	128	v7.3
ant17	2400 (P-IV)	512	v7.3

본 논문에서 사용되는 벤치마크는 다양한 형태의 계산 및 통신 방식을 가지는 프로그램들이다. 각 벤치마크에 대한 설명을 표 3에 종합하였다.

공정한 비교를 위해, 각 벤치마크에는 같은 최적화 옵션으로 컴파일된 동일한 core 코드를 사용하였다. 또한 실행 시간 측정은 프로그램의 core 부분과 work의 생성, 데이터 교환을 위한 부분만을 측정하였다. 프로그램의 환경 및 데이터 초기화에 필요한 시간 등은 실행 시간에 포함되지 않았다. 외부의 영향을 최소화하기 위해 실험이 진행되는 동안에는 다른 프로그램이 수행되지 않도록 하였으며, 동일한 실험을 다섯 번 반복하여 그 평균을 취하였다.

표 3. 실험에 사용된 벤치마크 설명

벤치마크	크기	설명
FIB	n=43	피보나치 수열 계산 알고리즘 $F(n) = F(n-1) + F(n-2)$
TSP	n=13	n개의 도시를 중복없이 모두 방문하는데 소요되는 비용 또는 시간을 산출하기 위한 알고리즘
NQ	n=14 (n*n)	n×n 체스판 위에 n개의 queen을 안전하게 놓을 수 있는 방법이 몇 가지인가를 탐색하는 알고리즘
MM	n=1100 (n*n)	매트릭스 두 개를 생성하고 그 곱을 계산하는 알고리즘

4.2 실험 결과

그림 2에 이기종 클러스터 시스템에서의 벤치마크 프로그램들의 성능 측정 결과를 종합하여 나타내었다. 그래프의 X축은 클러스터 시스템을 구성하는 컴퓨터의 수를 나타내었고, 주 Y축은 실행 시간을 초 단위로 표시하였으며, 보조 Y축에는 speedup을 나타내었다.

전체적으로 Cilk의 경우 모든 벤치마크 프로그램들이 이기종 클러스터 시스템의 성능을 적절히 활용하고 있는 것을 알 수 있다. Cilk의 dataflow 모델은 이기종 클러스터 시스템을 적절히 활용하고, 최적의 성능에 근접한 결과를 낼 수 있도록 하고 있다. WPM을 적용한 MPI 벤치마크 프로그램들의 경우 이기종 클러스터 시스템의 성능을 적절히 활용하는 Cilk의 결과에 근접한 결과를 보여준다. 그러나 여전히 Cilk보다 조금 못한 성능을 보여주고 있는데, 이는 WPM이 Cilk와 같이 완전하게 동적으로 동작하는 것이 아닌 정적으로 동작하는 부분이 남아 있기 때문이다. 그러나 WPM을 적용하지 않은 MPI 벤치마크 프로그램들에 비해 그 성능이 확실하게 증가되었다는 것을 확인 할 수 있으며, WPM을 적용함으로서 이기종 클러스터 시스템을 보다 적절히 활용할 수 있다는 것을 알 수 있다. Cilk와 WPM을 적용한 MPI의 성능 차이는 WPM이 동적으로 완전하게 동작하게 함으로서 극복할 수 있다.

Ⅴ. 결론

클러스터 시스템은 상업적으로 판매되는 개인용 컴퓨터들과 네트워크 장비들로 쉽게 구축할 수 있다. 급속도로 발전하고 있는 기술로 인해 클러스터 시스템을

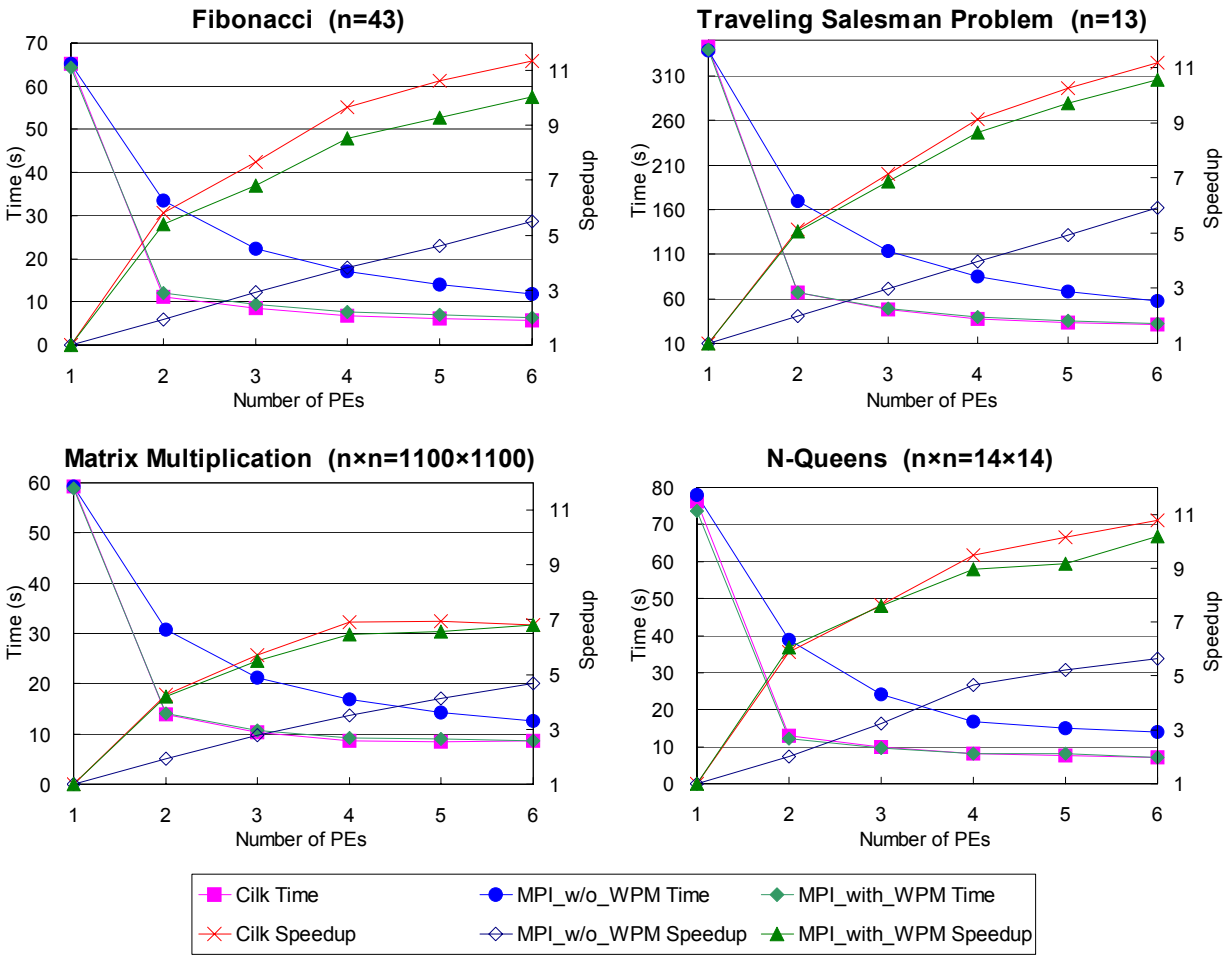

그림 2. 실험 결과

구성하고 있는 개별 컴퓨터를 교체하고 새로운 컴퓨터를 추가함으로서 전체 클러스터 시스템의 이기종화를 초래했다. 본 논문에서는 이기종 클러스터 시스템에서의 개별 컴퓨터를 효율적으로 활용하기 위한 방안에 대해서 살펴보았다.

본 논문에서는 이기종 클러스터 시스템의 성능을 최대한 활용할 수 있도록 하기 위해 Work Packet Manager (WPM)을 제안하였다. WPM은 MPI 프로그램들에 쉽게 적용할 수 있으며, 확장성 또한 제공하고 있다. 실험 결과에 따르면 WPM을 적용한 MPI 벤치마크 프로그램들의 성능이 이기종 클러스터 시스템을 효율적으로 활용하고 있음을 알 수 있다. MPI 병렬처리 프로그램을 개발함에 있어 WPM을 적용함으로서 이기종 클러스터 시스템의 성능을 효율적으로 사용할 수 있음을 증명하였다.

참고문헌

[1] M. Snir, MPI: The complete reference, MIT Press, 1996

[2] S. Baek, K. Lee, J. Kim, and J. Morris, "Heterogeneous Network of Workstations", Lecture Note on Computer Science, Vol. 3189, 2004, pp. 426-439.

[3] R.D. Blumofe, C.F. Joerg, B.C. Kuszmaul, C.E. Leiserson, K.H. Randall, and Y. Zhou, "Cilk: an efficient multithreaded runtime system", PPoPP'95, SantaBarbara, 1995

[4] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, "A high-performance, portable implementation of the MPI message passing interface standard", Parallel Computing, vol. 22, no. 6, pp. 789-828, Sep 1996.

[5] R. Buyya, "High Performance Cluster Computing - Architecture and Systems", Prentice Hall PTR, 1999