한국컴퓨터정보학회 동계학술대회 논문집 제22권 제1호 (2014. 1)

라즈베리 파이 클러스터 환경에서의 병렬 프로그래밍

정용재°, 김동관*
°*목포해양대학교 해양컴퓨터공학과
e-mail: mymy2401@naver.com°, dongkwan@gmail.com*

Parallel Programming on a Raspberry Pi Cluster

Yong Jae Jung°, Dong Kwan Kim*
°*Dept. of Computer Engineering, Mokpo National Maritime University

● 요 약 ●

일반적으로 컴퓨터 클러스터는 제공하는 기능에 따라 구축 비용이 고가이며 공간 확보, 냉난방, 전원 등을 위한 유지 비용을 무시할 수 없다. 그러므로, 일반 개발자들이 컴퓨터 클러스터를 보유하는 것은 쉬운 일이 아니다. 이러한 상황에서 교육용 및 연구용으로 가격대비 성능이 만족스러운 라즈베리 파이를 기반으로 한 컴퓨터 클러스터를 구축하여 분산 컴퓨팅 및 클라우드 컴퓨팅 시스템 개발을 위한 테스트 베드로 활용하고자 한다. 본 논문의 실험 결과는 라즈베리 파이 클러스터가 병렬 프로그래밍을 위한 효과적인 테스트 베드가 될 수 있음을 보인다.

키워드: 병렬 프로그램(parallel programming), Message Passing Interface (MPI), Raspberry Pi

I. 서 론[10)]

자연과학 및 공학 분야를 비롯하여 최근에는 빅 데이터 처리를 위한 병렬 및 분산 컴퓨팅에 대한 이론 및 시스템들이 개발되고 있으며, 고성능 컴퓨팅(High-Performance Computing, HPC)에 대한 연구가 활발히 진행되고 있다. 대형 슈퍼컴퓨터에서부터 데스크 탑 PC를 연결한 클러스터에 이르기까지 다양한 형태의 고성능 컴퓨팅 개발 및 운영 환경이 존재하며, 이를 기반으로 게임, 교육, 의료, 비즈니스 등 다양한 모바일 클라우드 서비스가 제공된다 [1].

매년 2회에 걸쳐 세계 슈터컴퓨터 순위를 발표하는 TOP500 Supercomputer 웹사이트 [2]에 의하면, 2013년 11월 현재 1위인 슈퍼컴퓨터는 3,120,000 개의 코어로 구성되고, 33.86 petaflop/s의 계산 능력을 보유하고 있다. 하지만, 해당 슈터컴퓨터를 유지하기 위해 17,808 kW의 전력을 소비되고 있다는 점에서 슈퍼컴퓨터 구축 및 활용 비용이 큰 부담이 되고 있다. 그러므로, 클라우드 컴퓨팅 서비스를 비롯하여 다양한 분야에 컴퓨터 클러스터들이 가격 대비 고성능의 컴퓨팅 능력을 제공함에도 불구하고, 여전히 개인이나 중소기업의 개발자들이 클러스터를 보유하기에는 용이하지 않다.

비용 및 공간 확보 측면에서, 라즈베리 파이 [3]와 같은 싱글 보드 컴퓨터의 보급은 컴퓨터 클러스터 구축을 위한 새로운 방향을 제시한다. 라즈베리 파이는 영국의 라즈베리 파이 재단이 기초 컴퓨터 과학 교육을 증진시키기 위해 만든 명함 크기의 싱글 보드 컴퓨터로 512MB의 RAM과 SD 카드를 외부 기억 장치로 사용하며 워드 프로세싱, 게임, 인터넷 브라우징 등이 가능하다.

본 논문에서는 라즈베리 파이 컴퓨터를 연결한 컴퓨터 클러스터 [4, 5]가 병렬 및 분산 컴퓨팅 연구를 위한 적절한 테스트 베드가 될 수 있음을 확인한다.

10) 이 논문은 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2012R1A1A1006022).

II. 본 론

그림 1. 실험에 사용된 라즈베리 파이 클러스터
Fig. 1. The Raspberry Pi Cluster used for the Experiment

본 논문에서 제시한 실험환경은 그림 1과 같다. 라즈베리 파이 컴퓨터 10대를 연결하여 실험용 컴퓨터 클러스터를 구성하며 한 개의 노드가 마스터 노드 기능을, 나머지 9개 노드가 작업 노드 기능을 수행한다. 각 클러스터 노드들은

ARM 프로세서와 512MB의 RAM으로 구성되며, 라즈베리 파

이 하드웨어에 최적화된 데비안을 기반으로 한 라즈비안(Raspbian) 운영체제에서 작동한다.

표 1. 소수 구하는 MPI 코드
Table 1. The MPI code for Finding Prime Numbers

```
//rank 0 노드의 작업 내용
if (rank == FIRST) {
    for (n=mystart; n<=LIMIT; n=n+stride) {
        if (isprime(n)) {
            pc++;  foundone = n;
            }
        }
MPI_Reduce(&pc,&pcsum,1,MPI_INT,MPI_SUM,FIRST,
MPI_COMM_WORLD);
MPI_Reduce(&foundone,&maxprime,1,MPI_INT,MPI_M
AX,FIRST,MPI_COMM_WORLD);
} /* End Of if (rank == FIRST) */

//rank 0 노드이외의 작업 내용
if (rank > FIRST) {
    for (n=mystart; n<=LIMIT; n=n+stride) {
        if (isprime(n)) {
            pc++;  foundone = n;
            }
        }
MPI_Reduce(&pc,&pcsum,1,MPI_INT,MPI_SUM,FIRST,
MPI_COMM_WORLD);
MPI_Reduce(&foundone,&maxprime,1,MPI_INT,MPI_M
AX,FIRST,MPI_COMM_WORLD);
} /* End Of if (rank > FIRST) */
```

라즈베리 파이 클러스터에 C 언어로 구현된 Message Passing Interface (MPI) [6] 병렬 프로그램 실행하여 병렬 프로그램을 위한 테스트 베드로 활용 가능성을 테스트 한다. 표 1은 실험에서 사용한 MPI 프로그램의 주요 소스 코드이다. 2,500,000까지의 숫자 중에서 소수(Prime Number)를 구하는 MPI 프로그램을 클러스터 노드 수를 2, 4, 8개로 변경하면서 실행 시간을 측정 비교한다. 또한, MPI 병렬 프로그램과 순차 프로그램의 실행 시간을 비교 측정하여 제시한다. 이는 병렬 프로그램이 순차 프로그램에 비해 실행 속도 면에서 장점이 있는지 확인하고자 수행한다. 실험결과는 그림 2와 같으며, X축은 프로그램의 유형 및 클러스터 노드 수를 나타내고, Y축은 실행 시간을 나타낸다. 병렬 프로그램이 순차 프로그램에 비해 소수를 찾는 실행 속도가 빠름을 알 수 있다. 또한, 연산에 참여하는 클러스터 노드의 수가 증가 할수록 실행 속도가 빨라짐을 알 수 있다.

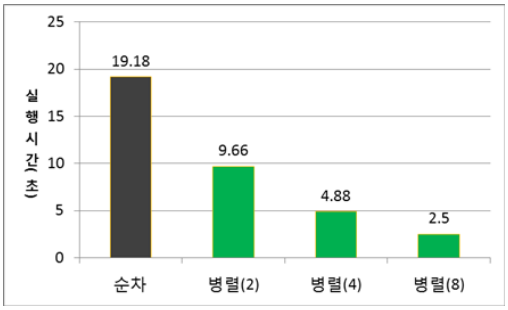

그림 2. 소수 구하는 MPI 프로그램 실행 결과
Fig. 2. Execution Results of the MPI Program for Finding Prime Numbers

III. 결 론

데스크 탑 PC의 저가화 및 고성능화로 인해 이를 연결한 클러스터들이 보급되고 있지만, 여전히 비용 및 공간 확보 측면에서 클러스터 구축 및 활용이 용이하지 않다. 이러한 상황에서, 라즈베리 파이 컴퓨터의 빠른 보급은 컴퓨터 클러스터의 대중화를 유발하고 있다. 본 논문은 라즈베리 파이를 이용하여 컴퓨터 클러스터를 구축하고, MPI 병렬 프로그램을 수행하여 병렬 컴퓨팅을 위한 테스트 베드로 활용한다. 본 논문의 실험에서 다양한 프로그램들이 테스트 되지는 않았지만, 실험 결과는 라즈베리 파이 클러스터가 병렬 프로그래밍을 위한 효과적인 테스트 베드가 될 수 있음을 보여준다.

참고문헌

[1] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, and Rajkumar Buyya, “Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges,” IEEE Communications Surveys & Tutorials January 2013.

[2] TOP500 Supercomputer, *http://www.top500.org/*.

[3] The Raspberry Pi Official Homepage, *http://www.raspberrypi. org/*.

[4] Joshua Kiepert, "Creating a Raspberry Pi-Based Beowulf Cluster," Boise State University, May 2013.

[5] Simon Cox, "Steps to make Raspberry Pi Supercomputer," University of Southampton, 2013, *http://www.southampton.ac.uk/~sjc/raspberrypi/pi_supercomputer_southampton.htm.*

[6] The Message Passing Interface (MPI) standard, *http://www.mcs.anl.gov/research/projects/mpi/*.