PLinda에서 master 선택과 성능과의 관계에 관한 연구

*홍석용 박영환
한성대학교 컴퓨터 공학과
phantom@hansung.ac.kr, yhpark@hansung.ac.kr

A Research for Relation between Master Selection and Performance in PLinda

Seokyong Hong Younghwan Park
Dept. of Computer Science Hansung University

요 약

PLinda는 기존 Linda의 개념에 폴트 톨러런트 개념을 추가하여 여러 대의 PC로 병렬 컴퓨터를 구성하는 방법을 제공한다. PLinda를 이용한 응용 중에 master-worker 구조의 것이 많은데, 이 때 master는 " job" 튜플을 만들어 튜플 스페이스에 저장하고 결과를 읽어서 종합하게 된다. 하지만 현재 다양한 성능을 보유한 PC들 중 어떤 것을 master로 선택해야 하는지에 대한 연구가 부족한 상황이다. 본 논문에서는 PLinda master를 결정함에 있어서 최고의 컴퓨팅 파워를 얻을 수 있는 방법의 모델을 제시하고, 수학적 검증을 하였다. 그리고 실제 병렬 시스템에서의 타당성을 확인하기 위해 실험적 검증도 하였다.

1. 서 론

PC의 급속한 성능 향상과 컴퓨터 네트워크의 급속한 발전은 슈퍼컴퓨터의 컴퓨팅 파워를 필요로 하는 응용프로그램을 값이 싼 PC들로 구성된 병렬 컴퓨터의 구성을 통하여 해결할 수 있는 여러 가지 길을 우리에게 제공하고 있다. 이러한 여러 가지 방법 중에 많이 쓰이는 개념들과 도구들로는 PVM(Parallel Virtual Machine)이나 MPI, Linda등과 최근에 대두되기 시작한 Grid Computing이 존재한다[6][7][4][8].

요즘 대부분의 대학들에는 적어도 3~4개의 PC 실습실을 보유하고 있는데, 이러한 PC들을 이용하여 대규모의 병렬 시스템을 구성할 수 있다. 이렇게 여러 대의 PC를 이용한 병렬 컴퓨터의 구성에 있어 문제점은 PC 시스템들이 동일하지 않을 수도 있다는 것이다. 즉 master-worker 패러다임에서 " job" 튜플을 생성하는 master 시스템과 컴퓨팅 파워만을 제공하는 worker 시스템을 어떻게 선택하느냐 하는 문제도 전체적인 병렬 컴퓨터의 성능에 영향을 줄 수 있다.

따라서 본 연구에서는 폴트 톨러런트 개념을 LINDA에 도입한 PLinda를 이용하여 이질적인(heterogeneous) 시스템으로 구성된 병렬 컴퓨터를 이용해 master-worker 구조의 처리를 할 때, 어떻게 master 시스템을 선택하는 것이 시스템의 성능 향상에 도움이 되는지를 하나의 모델을 제시하고, 이 모델의 타당성을 증명한 후 실험을 통하여 다시 한번 모델의 유효성을 검증하고자 한다

2. 본 론

2.1 Master 선택을 위한 모델

Master-worker 구조의 PLinda 시스템에서 최적의 수행시간을 얻어내기 위하여 master를 CPU 클럭 속도가 제일 낮은 PC에 실행하는 것이 효과적이다. 병렬 시스템에서 master-worker 응용은 CPU-bound이기 때문에 CPU 클럭 속도는 수행 시간에 막대한 영향을 미치기 때문이다. Master는 worker가 처리한 데이터를 바탕으로 최종의 결과만을 도출하고, master의 데이터 처리 작업은 worker에 비해 상대적으로 적기 때문이다. 물론 PLinda 응용 프로그램을 수행시킬 모든 PC의 CPU 성능이 동일하다면 master 설정에 큰 어려움이 없겠지만 각기 다른 H/W를 보유한 일반적인 단체의 경우 PLinda 시스템의 성능 향상을 위하여 이와 같은 master의 설정이 필요하다.

2.2 수학적 검증

병렬 처리 중 master-worker 구조의 것은 tuple space에 있는 job 튜플을 읽어 계산을 수행하는 작업이 대부분이다. 따라서 CPU-bound된 경향을 보이기 때문에 대부분의 경우 CPU의 성능이 전체적인 수행 속도에 매우 큰 영향을 미친다.

PLinda 프로그램의 총 수행 시간은 master와 worker 중 제일 긴 수행 시간이 된다.

$$PT_{total} = \max(PT_w, PT_m) \quad - (2)$$

Master 프로세스는 worker 프로세스들이 필요한 모든 데이터를 생성할 때까지 계속 수행을 하고, 단지 그 데이터를 수집하여 결과를 보여주기 때문에 다음과 같은 수식이 성립된다.

$$PT_m = PT_w \quad - (3)$$

따라서 총 수행 시간은 다음과 같다.

$$PT_{total} = PT_w \quad - (4)$$

수행 시간은 총 명령어의 수를 CPU 클럭 속도로 나눈 값이라고 가정하면 다음의 수식으로 표현할 수 있다.

$$PT = \frac{C(Number\_of\_Commands)}{V(CPU\_Clock\_Speed)} \quad - (5)$$

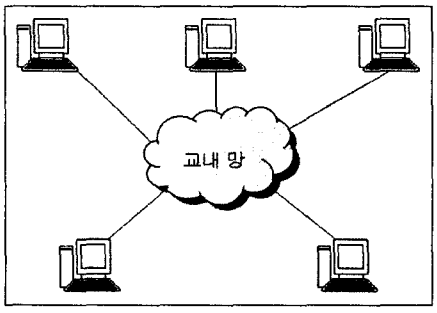
따라서 총 수행 시간은

$$PT_{total} = PT_w = \frac{C_w}{V_w} \quad - (6)$$

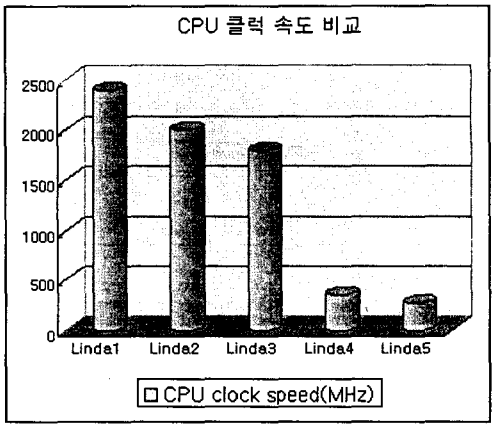
과 같다. worker가 처리하는 한 개의 job 크기가 일정하기 때문에 결국 총 수행 시간을 낮추기 위해서는 worker 프로세스를 CPU 클럭 속도가 빠른 PC에 배정해야 하고 결국 master 프로세스는 CPU 클럭 속도가 제일 느린 PC에 할당을 해야 한다.

2.3 유효성 검증을 위한 실험

모델의 유효성을 검증하기 위해서 다양한 성능을 갖고 있는 PC 들을 이용하여 병렬 컴퓨터를 아래 그림과 같이 구성하였다.



<그림 1. 실험 환경 구성>



<그림 2. 시스템 CPU 클럭 수 비교>

위 그림과 같이 구성된 병렬 컴퓨터에서 모델의 유효성을 검증하기 위하여 만델브로트 셋(Mandelbrot Set)을 응용 프로그램으로 선택하여, 이를 PLinda 개념을 통하여 병렬화 하였다. Master는 튜플-스페이스에 튜플 그룹을 생성하고 worker수만큼 프로세스를 만든다. Worker 는 프로세스를 지명을 받고 master가 쓴 " job" 튜플을 읽고 지운다. 다음 프로세스가 똑같은 튜플에 접근하지 못하게 하기 위함이다. worker에서는 계산루틴을 다 돌고 " done" 이라는 키워드로 튜플을 생성한다. master는 pl_in() 프리미티브를 호출하여 블록 되어 있다가 " done" 이라는 튜플이 생성되면 읽고 지운다. 이 같은 절차를 worker가 끝날 때까지 반복 처리하게 된다.

2.4 실험 결과

실험에 사용한 5대의 PC 각각에 하나의 master를 수행하고 master를 제외한 나머지 PC에 하나씩의 worker를 수행하여 수행이 완료되는 시간까지 걸린 시간을 측정하였다. 다음의 표-1은 그 측정 결과값이다.

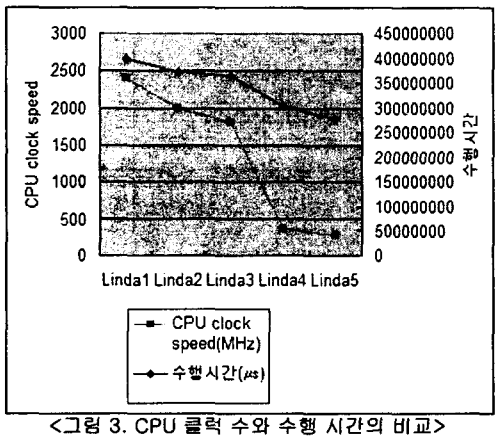
Linda1	Linda2	Linda3	Linda4	Linda5
399.78	374.47	364.66	303.27	279.66

표-1. 수행 시간 (단위 : s)

CPU 클럭 속도가 가장 좋은 Linda1에 master를 수행시킨 경우 만델브로트 셋을 완성하는데 399.77초 걸린 반면, 클럭 속도가 제일 낮은 Linda5에 master를 수행시킨 경우 279.66초 걸린 것을 확인할 수 있다.

위의 그래프를 보면 master를 수행하는 PC의 CPU 클럭 속도가 낮을수록 만델브로트 셋의 완성 시간이 크게 감소하는 것을 확인할 수 있다. 특히 CPU 클럭 속도가 많이

차이 나는 Linda3과 Linda4의 경우 수행 시간의 차이가 크게 발생한다. 이 같은 결과가 나오는 것은 master-worker의 특성에 기인한 점이 크다. 우선 master는 처리할 " job" 튜플을 생성, 분배하고 worker에 의해 처리된 " done" 튜플을 읽어 화면에 표시하는 것이 대부분이다. 반면 worker는 " job" 튜플을 읽어서 실제로 해당 영역에 맞는 그래픽 처리를 위한 계산 작업을 수행한다. 그리고 그러한 " job" 들이 많기 때문에 master보다 더 많은 CPU 자원을 필요로 하게 된다. 따라서 worker의 처리 속도가 전체적인 수행 속도에 많은 영향을 미치게 된다. 그렇기 때문에 CPU 사용량이 적은 master 프로세스를 CPU 클럭 속도가 낮은 PC에 할당하고, 상대적으로 CPU 사용량이 많은 worker 프로세스를 CPU 클럭 속도가 높은 PC에 할당하는 것이 전체적인 수행 속도를 최적화할 수 있는 배치 방법임을 알 수 있다.



<그림 3. CPU 클럭 수와 수행 시간의 비교>

3. 결론 및 향후 연구 과제

앞의 실험을 통하여 볼 수 있듯이 CPU의 클럭 속도에 따라서 PLinda 응용프로그램은 큰 성능 차이를 보였다. 이는 master-worker 응용이 CPU-bound이고, worker가 master보다 상대적으로 더 많은 연산을 해야 하기 때문이다. 따라서 연산이 많은 worker 프로세스에 CPU 클럭 속도가 더 높은 PC를 할당하는 것이 전체적인 PLINDA 응용 프로그램의 속도를 향상시킬 수 있는 방법임을 알 수 있었다.

본 논문에서 우리는 PLinda 시스템을 이용한 병렬 처리에 있어서 다양한 하드웨어를 보유한 PC들 사이에 어떤 것을 master로 정해야 전체적인 성능 향상을 가져올 수 있는지에 대하여 모델을 제시하고 수학적 증명을 시도하였으며, 실제적으로 그런 모델이 같은 결과를 보이는지 실험을 통하여 검증하였다. 우리가 실험을 수행한 환경은 교내 네트워크에 적은 개수의 PC로 구성된 PLinda 시스템이다. 본 논문에서 제시한 모델이 실험 환경에서 정확하게 검증이 되었다. 하지만 실제 학교 같은 조직에서는 더 많은 PC들이 존재하고 많은 시간 동안 컴퓨팅 파워를 낭비하고 있는 것이 현실이다. 따라서 차후에는 더 많은 PC들로 실험을 하여 우리가 제시한 모델이 성립하는가에 대한 연구가 요구된다. 또 극심하게 변하는 네트워크 환경에서 어떤 결과를 보이는지도 연구할 필요가 있다. 병렬 처리를 위해 여러 대의 PC로 시스템을 구성하는 PLinda는 실제로 다양한 H/W 및 운영 체제가 존재하는 환경에서도 동작을 해야 하지만 아직까지는 다양한 운영 체제 환경을 지원하지 못하는 것이 사실이다. 따라서 실험도 리눅스 기반의 환경에서 수행하였다. 하지만 실제로 국내의 경우 리눅스 보다는 윈도우 환경의 PC가 많은 상태이다. 더 많은 가용 자원을 병렬 시스템화 하기 위해서 다양한 환경을 지원하는 PLinda에 대한 연구가 중요하다. 그리고 아직까지는 PLinda 응용 프로그램을 일일이 각각의 PC에 설치해야 하는 번거로움을 줄여 나가기 위한 연구가 필요하다.

참고 문헌

[1] 박영환, 병렬 프로그래밍 개념 LINDA의 간편화 , 한국정보과학회논문집, 2000

[2] 박영환, 시간성 튜플 도입을 통한 LINDA 개념의 확장 , 한국정보과학회논문집, 2001

[3] Karpjoo Jeong, "An Approach to Fault-tolerant Parallel Processing on Intermittently Idle, Heterogeneous Workstations"

[4] Nicholas Carriero and David Gelernter, "LINDA IN CONTEXT", Communications of the ACM, 1989

[5] S.Ahuja, N.Carriero, and D.Gelernter, "Linda and Friends", Computer, Vol. 19, No. 8, Aug. 1986.

[6] Richard A. Sevenich, " Parallel Processing using PVM" , " Linux Journal" , 1998

[7] CORPORATE The MPI Forum, " MPI:a message passing interface" , " Proceedings of the 1993 ACM/IEEE conference on Supercomputing" , 1993

[8] Henri Casanova, " Distributed computing research issues in grid computing" , " ACM SIGACT News" , 2002

[9] Karpjoo Jeong and Dennis Shasha, " PLinda 2.0: A Transactional/Checkpointing Approach to Fault Tolerant Linda" ,Proc. the 13th International Symposium on Reliable Distributed Systems. Oct. 1994

[10] Karpjoo Jeong, Bin Li, Dennis Shasha and Peter Wyckoff, " PLinda/C++ and Fortran77 User' s Guide" , 1996

[11] 박영환, " 실시간 병렬 커널 : PRT-LINDA의 설계 및 구현" , " 한성대학교" , " 정보통신연구소 정보통신논문집 3권, 29-39pp"