

# 다중 에이전트 시스템 기반의 병렬 계산을 위한 작업 할당 기법과 성능비교

김경하<sup>U</sup>                      김영학                      오길호  
금오공과대학교 컴퓨터공학과  
(khkim, yhkim, gilho)@cespc1.kumoh.ac.kr

## Task Allocation Methods and Performance Comparison for Parallel Computation Based on Multi-Agent System

Kyong-Ha Kim<sup>U</sup>      Young-Hak Kim      Gil-Ho Oh  
Dept. of Computer Engineering, Kumoh National University of Technology

### 요 약

최근 높은 컴퓨팅 파워를 요구하는 응용문제 처리를 위해 고 비용의 슈퍼컴퓨터 대신 인터넷 상에 분산된 다수의 일반 컴퓨터들을 이용하는 병렬처리에 관한 연구가 진행되고 있다. 본 연구에서는 다중 에이전트 시스템을 기반으로 하여 이질적인 성능을 갖는 컴퓨터들을 병렬 컴퓨팅 환경으로 구성하고 각 호스트의 성능측정 결과에 따라 효율적으로 작업을 분산하는 기법을 제안한다. 또한 본 연구에서 제안한 방법을 다중 에이전트 시스템인 IBM의 Aglets<sup>TM</sup>을 사용하여 실험적으로 성능을 평가하고, 기존 연구와 비교한다.

### 1. 서론

최근에 높은 컴퓨팅 파워를 요구하는 응용 분야에 고 비용의 슈퍼컴퓨터를 대신하여, 네트워크 상에 분산되어 있는 유휴 자원을 이용해 문제를 해결하기 위한 다양한 연구가 진행되고 있다[1,2,3,4]. 다중 에이전트 시스템을 이용하여 병렬 컴퓨팅 환경을 구성하려는 연구도 이들 분야중의 하나이다. 이러한 다중 에이전트 환경에서는 네트워크 상에 분산되어 존재하는 서로 다른 성능을 갖는 컴퓨터들을 하나의 가상 병렬 컴퓨팅 환경으로 구성하고, 주어진 문제에 대한 하나의 큰 작업을 분할하여 다수의 컴퓨터 상에서 효율적으로 분산처리 할 수 있다[4,5,6].

우리는 [1]에서 동일한 컴퓨터를 갖는 가상 병렬 컴퓨팅 환경을 구성하고 워커 에이전트(이하 WA)와 작업 패키지를 효율적으로 분산하는 방법을 제안하였다. 그러나 인터넷의 실제 환경에 존재하는 각 컴퓨터는 서로 다른 작업 처리 능력을 갖는다. 본 논문은 이러한 일반적인 환경을 고려하여 구성된 가상 병렬 컴퓨팅 환경에서 각 호스트의 처리능력에 따라 작업을 분산하는 방법을 제안하고, 실험적인 결과를 통하여 이미 알려진 방법들과의 성능을 비교한다.

본 논문에서 가상 병렬 컴퓨팅 환경을 구성하기 위해 IBM의 Aglets<sup>TM</sup> 시스템을 사용하였다.

### 2. 관련 연구

인터넷 상의 병렬 컴퓨팅은 크게 메시지 전달(Message Passing;이하 MP) 모델과 분산 공유 메모리(Distributed Shared Memory;이하 DSM) 모델로 구분하여 연구되고 있으며, 최근에 에이전트 연구가 진행되면서 이동성을 가진 에이전트의 장점을 이용한 다중 에이전트 시스템(Multi-Agent System) 기반의 병렬 컴퓨팅 연구 또한 활발히 진행되고 있다. MAS 기반의 병렬 컴퓨팅 환경은 다른 방법과 비교했을 때, 코드와 데이터가 계산하는 호스트로 이동하기 때문에 전체적인 통신

회수를 줄여 네트워크 부하를 감소시킬 수 있고 시스템 구성이 편리하다는 장점을 갖는다. 또한 MP 방식과 DSM 방식은 관련된 프로그램이 수정 될 때마다 병렬 컴퓨팅에 참여하는 모든 호스트에 새로 설치해야 한다. 그러나 이동성을 가진 MAS는 초기 한번만 각 호스트에 에이전트 서버를 설치하면 되고, 실제 계산은 에이전트가 이동하면서 동적으로 수행한다. 다음은 MAS를 기반으로 하여 작업을 분산하는 기존 방법들을 개략적으로 소개한다.

#### 2.1 코디네이터를 이용한 방법

병렬 컴퓨팅 환경에서 하나의 컴퓨터가 코디네이터 역할을 하게되며, 이 코디네이터가 다수의 호스트에 분산 작업을 수행하는 WA를 배치한다. 그리고 주어진 응용문제를 여러 개의 단위 작업으로 WA에게 분산하고 다시 결과를 수집한다[4]. 여기서 병렬성을 가진 하나의 문제가 주어졌을 때 코디네이터는 각 WA에게 균일한 크기로 나누어진 단위 작업을 보내고, 각 WA로부터 결과를 반환 받게 된다. 이러한 작업은 코디네이터가 더 이상의 단위 작업을 갖지 않을 때까지 반복한다. 이 방법의 문제점은 첫째, 작업의 양에 따라 성능 향상을 위해 WA 수를 일정 범위 이상 증가시키면 오히려 성능이 떨어지는 단점이 있다. 둘째, 코디네이터의 분산 수행시간이 각 WA의 평균 수행시간보다 더 길어지면 코디네이터에 부하가 집중되어 병목현상이 발생하게 된다.

#### 2.2 마스터/슬레이브 방법

위에서 설명한 것과 비슷한 방법으로 이 방법에서 병렬 컴퓨팅 환경은 하나의 마스터와 다수의 슬레이브들로 구성된다[7]. 이 방법과 코디네이터를 이용한 방법[4]의 차이점은 다음과 같다. [4]의 방법은 병렬 컴퓨팅에 참여하는 호스트 수와 관계없이 주어진 작업을 균일한 크기의 단위 작업으로 분할하지만, 마스터/슬레이브 방법에서는 처음부터 컴퓨팅에 참여하는 슬레이브 수 만큼 작업을 균등하게 분할하여 작업을

분산한다. 그 결과 마스터에 집중된 부하를 어느 정도 감소시켜 전체적인 성능을 높일 수 있지만, 참여하는 호스트의 수가 많아지면 여전히 마스터에 부하가 집중되어 성능이 떨어진다.

2.3 위임에 의한 분산 방법

이 방법은 코디네이터에 집중된 부하를 분산하기 위해 각 WA가 코디네이터의 일부 역할 즉, 에이전트와 작업을 분산하는 것을 위임받아서 다수의 호스트가 동시에 작업을 분산에 참여하게 된다[1]. 그 결과 코디네이터의 부하를 WA들에게 분산함으로써 더 향상된 성능을 보였으나, 실제환경에서의 검증과 이질적 환경을 고려하지 않았다.

본 논문에서는 이러한 문제점들을 고려하여 [1]에서 제안된 방법을 기본으로 하여, 이질적 병렬 컴퓨팅 환경에 적용할 수 있는 새로운 분산방법을 제안하고 기존의 방법들과의 성능을 비교한다.

3. 이질 환경을 고려한 분산 기법

이질적인 호스트로 구성된 병렬 컴퓨팅 환경에서 각 WA들에게 작업을 동일하게 분할하여 수행하게 하면, 성능이 좋은 호스트의 WA인 경우 그 결과를 빨리 응답하지만, 성능이 좋지 않은 호스트의 WA는 더 늦게 응답하게 된다. 따라서 각 호스트들의 성능을 고려하여 적절한 작업 분배를 함으로서, 전체적 계산시간을 단축시킬 수 있다.

분산 기법에 대한 자세한 설명은 [1]을 참조하길 바라며, 그림 1과 그림 2는 8대의 호스트로 구성된 병렬 컴퓨팅 환경에서 WA와 작업 패키지 분산 그리고 결과를 수집하는 과정을 보여준다. 여기서 H<sub>0</sub>-H<sub>7</sub>은 에이전트와 분할된 작업이 이동하는 호스트이며, ①-③은 분산/통합 단계를 의미한다.

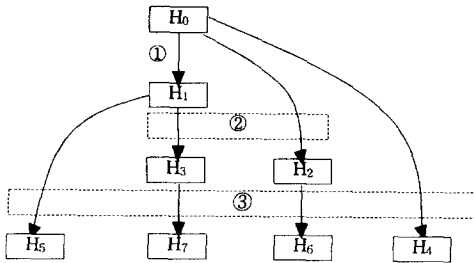


그림 1 에이전트와 작업 패키지 분산 방법

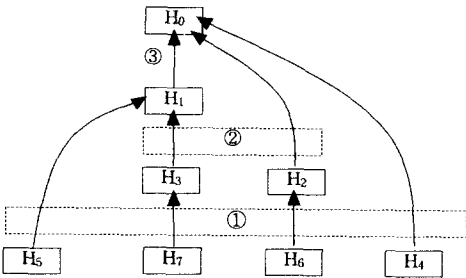


그림 2 결과 통합 방법

3.1 작업 패키지 분산 할당 기법

호스트에 WA를 배치시키기 전에 먼저 코디네이터가 성능 측정 에이전트를 각 호스트로 보내고, 각 호스트의 성능을 수집하여 각 호스트로 분할된 작업량을 결정한다. 각 호스트의 성능 평가는 여러 가지 기법이 있을 수 있으나, 본 논문에서는 수행시간이 짧고 간단한 방법으로 성능을 측정하는 Linpack 벤치마킹[8]을 적용한다. 이 벤치마킹 기법은 행렬을 이용한

선형방정식(Ax=b)의 해를 구하는 알고리즘을 적용하여 컴퓨터의 부동 소수점 연산능력을 평가한다. 방정식의 해는 부분 피벗팅을 이용한 가우스 소거법을 이용하며, 단위 시간당 부동 소수점 덧셈과 곱셈 연산의 수행 횟수를 검사하여, 초당 몇 백만번의 부동소수점 연산을 수행하는지에 대한 MFLOPS 단위의 값을 측정한다. 작업을 수행하는 호스트는 각 호스트의 결과를 받아서 성능이 좋은 순서로 ip\_table을 재구성하며 호스트의 성능 비를 계산해서 ip\_table에 저장한다. 각 호스트로 분산 될 작업은 이 성능 비로 결정된다. 표 1은 성능 비에 따른 작업 패키지 할당 정책을 설명하기 위한 기호이다.

기호	의미
$w_i$	$i$ 번째 워커 에이전트
$m$	병렬 컴퓨팅에 참여하는 호스트 수
$n_i$	$w_i$ 에 할당되는 작업 패키지량
$N$	전체 작업 패키지량
$c_i$	각 호스트의 성능(MFLOPS)
$c_{total}$	ip_table의 전체 호스트의 성능 합계
$r_i$	$i$ 번째 호스트의 성능비

표 1 작업 패키지 할당정책에 사용될 기호

$c_{total}$ 은 병렬 컴퓨팅에 참여하는 전체 호스트의 성능 합계로 다음 수식으로 구할 수 있다.

$$c_{total} = \sum_{i=0}^{m-1} c_i$$

$r_i$ 는 각 호스트에 할당할 작업 패키지량을 결정하는 성능 비이며, 각 호스트의 성능을 전체 호스트의 성능합계로 나누어 구할 수 있다. 계산된 각 호스트의  $r_i$ 는 ip\_table에 저장된다. 계산식은 다음과 같다.

$$r_i = \frac{c_i}{c_{total}}, (0 \leq r_i \leq 1)$$

$n_i$ 는 ip\_table에서  $i$ 번째 호스트에 할당되는 작업 패키지량을 의미하며 다음과 같다.

$$n_i = N \times r_i$$

WA는 자신이 거주하게 될 호스트를 포함한 하위의 호스트로 구성된 ip\_table을 가지고 이동하며  $r_i$ 와 자신이 할당받은 작업량으로 각 호스트에 할당할  $n_i$ 를 결정한다.

3.2 알고리즘

다음은 각 호스트의 성능을 측정하고 WA와 작업 패키지를 분산하는 개략적인 알고리즘을 보여준다. 각 호스트에서 수행된 결과를 통합하는 방법은 분산방법과 유사하며, 지면관계상 생략한다.

```

procedure PARALLEL( )
    Host_Performance(old_ip_table, measure_agent)
    Distribution(new_ip_table, worker_agent, package)
end procedure

procedure Host_Performance(old_ip_table, measure_agent)
    for i=0 to m-1
        measure_agent creation과 dispatch
    end for
    for i=0 to m-1
        Linpack benchmark를 수행하여 초당 부동소수점 연산
        (MFLOPS) 결과를 코디네이터로 전송
    end for
    new_ip_table = MFLOPS값의 내림차순으로 ip_table sorting
end procedure
    
```

```

procedure Distribution(new_ip_table, worker_agent, package)
  for i=0 to logm-1
    for j=0 to 2i-1 in parallel
      processor Pj
      1) worker_agent creation
      2) j번째 호스트가 2i+j 번째 호스트로
         worker_agent dispatch
      3) Task_Division(j, package)
      4) j번째 호스트의 package를 2i+j 번째 호스트로 전송
    end for
  end for
end procedure

procedure Task_Division(j, package)
  1)  $subtask = \frac{\sum_{x=0}^{m-1} r_{even,x}}{\sum_{x=0}^{m-1} r_x} \times n_j$ ;
     *  $\sum_{x=0}^{m-1} r_x$ 는 j번째 호스트에서 ip_table의 MFLOPS 합계
     *  $\sum_{x=0}^{m-1} r_{even,x}$ 는 j번째 호스트에서 ip_table의 짝수 번째
       호스트의 MFLOPS 합계
     *  $n_j$ 는 j번째 호스트로 전송된 작업량(=package)
  2) package = subtask
end procedure
    
```

4. 성능 평가

성능 평가를 위해 Solaris™ 및 Windows 98™ 운영체제를 가지며 10Mbps 이더넷으로 연결된 워크스테이션 및 PC 상에 IBM Aglets™을 설치하여 병렬 컴퓨팅 환경을 구성하였으며, 이러한 환경에서 행렬 곱셈 문제를 Java™로 구현하여 실험적인 성능 결과를 비교하였다.

4.1 동일 환경에서의 성능비교

먼저 모든 호스트가 각각 110MHz 프로세서와 32MB 메모리로 구성된 환경을 고려하였다. 그림 3은 워크스테이션의 수를 16대로 구성한 병렬 컴퓨팅 환경에서 행렬 곱셈 문제를 수행한 결과를 보여준다.

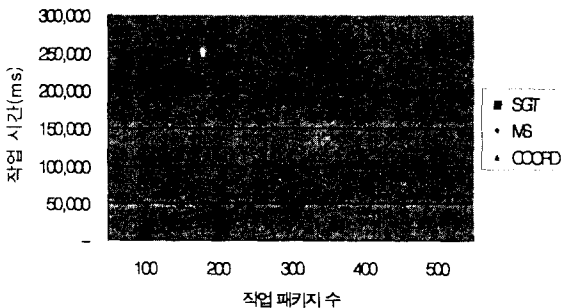


그림 3 동일 환경에서의 성능비교

여기서 SGT는 본 논문에서 제안한 방법이며, MS는 마스터/슬레이브 방법을 나타내고, COORD는 코디네이터 방법을 의미한다. 그림에서도 알 수 있듯이 동일 성능의 호스트로 구성되어 있을 경우, 본 논문에서 제안한 방법이 기존의 방법보다 더 향상된 성능을 나타냄을 알 수 있다. 이러한 결과는 작업 패키지당에 따라 호스트 수를 증가시키면 마스터/슬레이브 방

식이나 코디네이터 방식은 부하가 집중되지만, 제안한 방법은 WA로 부하를 분산시키기 때문에 더 좋은 결과를 얻을 수 있게 된다.

4.2 이질적 환경에서의 성능 비교

다음은 워크스테이션 2대(각 110MHz), 펜티엄 II 300MHz(64MB) PC 1대, 펜티엄 II 400MHz(128MB) PC 1대, 모두 4대로 실험환경으로 고려하였다. 이러한 환경에서 행렬 곱셈을 수행한 실험 결과는 그림 4와 같다.

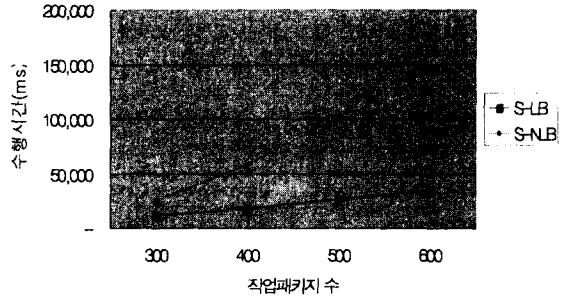


그림 4 이질적 환경에서의 성능비교

여기서 S-LB는 본 논문에서 제안한 방법이며, S-NLB는 이질 환경을 고려하지 않은 [1]에서 제안한 방법을 의미한다. 그림 4에서 보듯이와 같이 각 호스트의 성능 지수에 따라 작업을 할당할 때 그렇지 않은 경우보다 훨씬 더 좋은 성능임을 알 수 있다.

5. 결론

본 논문은 다중 에이전트 기반의 이질적인 병렬 컴퓨팅 환경을 고려하여 성능기반 작업 할당 기법을 제안하고 성능을 평가하였다. 실제 환경에서의 실험결과 제안한 방법이 다른 기존 방법에 비해서 성능이 더 향상되었음을 보였다.

참고문헌

- [1] 김경하, 김영균, 김영락, 오길호, "이동 에이전트 시스템 기반의 병렬 계산을 위한 효율적인 분산 방법", 한국정보과학회, 봄 학술논문집(A), pp.615-617, 2000.
- [2] 김중권 외 3인, "MPI 표준 기능 및 기술동향 고찰", 병렬 처리 시스템 연구회지 7권 1호, 1996.
- [3] L.F.G. Sarmenta, "Bayanihan: Web-based volunteer computing using java", <http://www.cag.lcs.mit.edu/~bayanihan>, 1998.
- [4] M.Straber, J.Baumann, M.Schwehm, "An agent-based framework for the transparent distribution of computations", PDPTA, vol.1, pp.376-382, 1999.
- [5] D. Chess, C. Harrison, A. Kershenbaum, "Mobile agents: are they a good idea?", 1995.
- [6] M. Starber, M. Schwehm, "A performance model for mobile agent systems", PDPTA, vol. II, pp.1132-1140, 1997.
- [7] D. B.Lange, M. Oshima, Programming and deploying java mobile agents with aglets, Addison wesley, 1998.
- [8] J. Dongarra, "Performance of various computers using standard linear equations software", <http://www.netlib.org/benchma/performance.ps>, 1998.