

모바일 그리드 환경에서의 작업 분배 기법†

정대용*, 진성호*, 정광식**, 유현창*, 길준민***

*고려대학교 컴퓨터교육학과,

**한국방송통신대학교 컴퓨터과학과,

***대구가톨릭대학교 컴퓨터정보통신공학부

e-mail:karat@korea.ac.kr

Task Distribution Mechanism in Mobile Grid Environments

Daeyong Jung*, SungHo Chin*, KwangSik Chung**, HeonChang Yu*, JoonMin Gil***

*Dept of Computer Science and Education, Korea University

**Dept. of Computer Science, Korea National Open University

***School of Computer & Information Communications Engineering, Catholic University of Daegu

요 약

모바일 그리드 환경에서 작업 처리를 위해 노드로 활용되는 모바일 장치는 무선 인터넷 연결과 모바일 장치의 배터리 용량 등의 제약으로 인해 모바일 그리드 환경에서의 작업 처리는 기존 그리드 컴퓨팅 환경에 비해 신뢰성이 낮고 결함이 발생할 확률이 높다. 따라서 모바일 장치의 신뢰성을 고려한 작업 분배 방법이 요구된다. 이 논문에서는 모바일 장치의 신뢰성을 고려한 작업 분배 스케줄링을 제안하고자 한다. 제안 기법에서는 노드의 신뢰성을 바탕으로 작업량과 복제 개수를 서로 달리 설정하여 신뢰성이 우수한 노드에게 더 많은 작업량을 할당하고 작업 복제 개수를 적게 설정하도록 한다. 제안 기법의 타당성을 보이기 위한 성능 평가에서는 제안 기법이 기존 기법에 비해 성능이 향상됨을 보여준다.

1. 서론

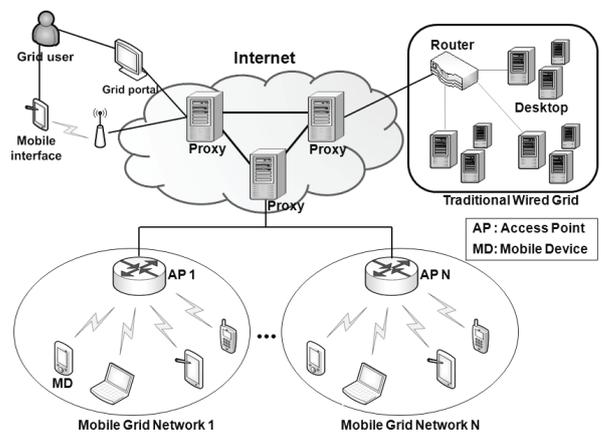
그리드 컴퓨팅은 지리적으로 분산된 컴퓨팅 자원을 가지고 대량의 데이터 처리나 복잡한 계산을 해결하기 위한 분산 컴퓨팅 패러다임이며, 복잡한 계산을 필요로 하는 생명과학, 우주과학, 기상예측 등 다양한 분야에서 적용되고 있다[1, 2]. 최근, 노트북, 스마트폰, PDA 등과 같은 무선 모바일 장치의 보급으로 이 장치들을 그리드 자원으로 이용하려는 연구가 활발히 이루어지고 있다. 그러나 모바일 장치를 이용한 모바일 그리드에서는 모바일 장치의 무선 신호세기, 배터리 용량과 같은 제약 사항을 필연적으로 가지고 있다. 이런 제약 사항으로 인해 작업 수행 중 결함 발생을 방지하기 위한 대표적인 해결책으로 작업 복제 기법이 전형적으로 활용되고 되고 있다[3]. 작업 복제 기법에서는 모바일 장치의 신뢰성에 따라 작업을 할당하고자 하였으며, 또한, 같은 작업을 여러 노드에 복제하여 동시에 작업을 수행함으로써 결함 예방을 할 뿐만 아니라 작업의 신뢰성도 높이고자 하였다.

그러나, 기존 연구에서는 일정한 신뢰성을 가진 노드들에게 같은 작업량을 분배하고 분배된 작업을 처리하여 모바일 그리드의 동적 환경에 적절하게 대처하기 어렵다.

그리고 노드의 신뢰성에 관계없이 일정한 작업 복제 개수를 정적으로 설정하여 모바일 그리드의 신뢰적인 성능을 보장하지 못하였다[4].

본 논문에서는 기존 그리드에 비해 작업 제약 사항이 많은 모바일 그리드 환경에서 효율적인 자원 분배를 위해 노드의 신뢰성을 고려한 작업 분배 기법을 제안한다. 제안 기법은 노드의 신뢰성에 따라 작업량을 서로 다르게 분배하여 모바일 그리드 환경의 동적 특성을 적절히 반영하도록 한다.

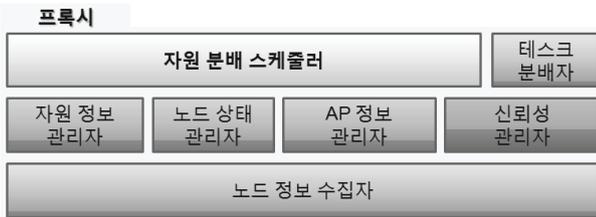
2. 시스템 구성



(그림 1) 시스템 구성

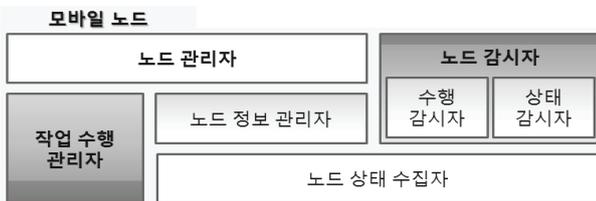
† 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (2009-0077638).

본 논문에서 제안한 시스템 구성은 (그림 1)과 같다. 이 그림은 유선 그리드 네트워크, 무선 그리드 네트워크를 프록시 서버로 연결한 형태로 구성한다. 프록시 서버는 모바일 장치의 정보와 AP의 정보를 가지고 작업을 할당하는 역할과 모바일 장치의 정보 상태 및 AP 상태를 관리하는 역할을 한다. 다음은 프록시와 노드의 구조에 대해서 설명한다.



(그림 2) 프록시 구조

프록시 서버의 구조는 (그림 2)와 같다. 이 그림에서 노드 정보 수집자는 모바일 장치의 정보를 수집하는 역할을 한다. 그리고 수집된 정보를 바탕으로 모바일 장치의 정보를 관리하는 자원 정보 관리자, 모바일 장치의 상태를 관리하는 노드 상태 관리자, AP의 정보를 관리하는 AP 정보 관리자, 그리고 노드의 자원 이용률과 상태를 관리하는 신뢰성 관리자가 있다. 이 4가지의 관리자를 통하여 자원 분배 스케줄러는 개별 노드의 작업량을 설정하고 태스크 분배자를 통하여 나누어진 작업들을 노드에게 할당하는 역할을 한다.



(그림 3) 모바일 노드 구조

모바일 노드의 구조는 (그림 3)과 같다. 이 구조에서는 노드의 자원 상태, 무선 신호 세기 상태, 배터리 상태를 노드 상태 수집자가 수집하고 수집된 정보를 노드 정보 관리자가 관리한다. 그리고 이 정보들은 노드 관리자를 통하여 프록시 서버에 있는 노드 정보 수집자에게 전송된다. 작업수행 관리자는 할당받은 작업의 정보를 관리하고 노드 감시자는 작업 상태 및 노드의 상태를 모니터링 한다.

3. 작업 분배 스케줄러

이 장에서는 효율적으로 작업을 모바일 노드에게 분배하는 스케줄러에 대해 설명한다.

우선, 노드의 신뢰성 r_i 는 다음과 같이 계산된다.

$$r_i = 1 - f_i \quad (1)$$

여기서 f_i 는 작업의 실패 확률을 나타낸다. 전체 모바일

노드에서 작업을 할당할 노드의 신뢰성은 95% 이상으로 보고 있다[4]. 그러므로 본 논문에서는 전체 노드 N 에서 작업 가능한 확률인 t 를 95%이상으로 설정한다. 그리고 신뢰성별 그룹을 나누기 위해 그룹 개수 N_g 를 다음과 같은 기준에 의해 설정한다.

$$N = \{m_i | r_i \geq t, 1 \leq i \leq n\} \quad (2)$$

$$N_g = \{m_i | T_{g-1} \geq r_i \geq T_g, 1 \leq i \leq n\} \quad (3)$$

다음으로 각 그룹 별로 작업을 분배하기 위해 필요한 그룹별 신뢰성과 그룹별 작업 복제 개수를 계산한다. 먼저 그룹 g 의 신뢰성 R_g 는 다음과 같이 계산된다.

$$R_g = 1 - \frac{\sum_{i \in N_g} f_i}{|N_g|} \quad (4)$$

여기서, N_g 는 그룹 g 를 나타낸다.

그룹별 작업 복제 개수 C_g 는 다음과 같이 정의된다.

$$C_g = \left\lfloor \frac{R_G}{R_g} \times C_G \right\rfloor \quad (5)$$

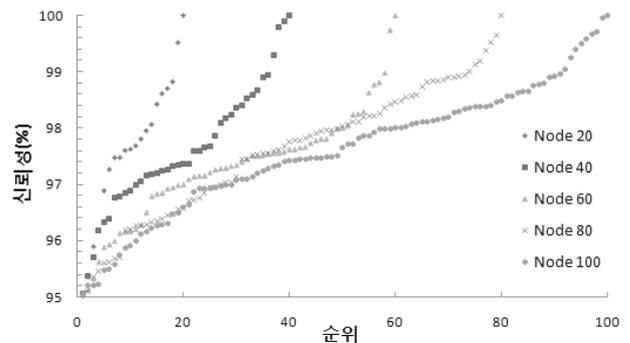
여기서, R_g 와 R_G 는 그룹 g 의 신뢰성과 전체 신뢰성을 각각 나타낸다. 그리고 C_G 는 사용자가 정의한 복제 개수를 의미하고 있다. R_g 와 C_g 를 이용하여 그룹별 작업량을 분배한다. 이때, 그룹별 작업량 J_g 는 다음과 같이 정의된다.

$$J_g = \frac{1}{2} \times \left(\frac{R_g}{\sum_{i=1}^G R_i} + \frac{\frac{|N_g|}{C_g}}{\sum_{i=1}^G \frac{|N_i|}{C_i}} \right) \times J_{total} \quad (6)$$

여기서, J_{total} 은 전체 작업량을 나타낸다.

4. 성능 평가

이 장에서는 제안한 작업 분배 스케줄러의 성능을 검증하기 위해 실험을 통해 성능을 평가한다. 제안한 기법은 모바일 노드의 신뢰성을 기반으로 작업을 분배하기 때문에 실험에 사용한 노드 개수별 신뢰성을 먼저 구하고 이 신뢰성을 바탕으로 성능을 비교한다. (그림 4)는 순위별로 정렬된 노드의 신뢰성을 보여준다.



(그림 4) 노드의 신뢰성

실험을 위해 본 논문에서 사용한 파라미터는 <표 1>과 같다. 이 파라미터 값을 이용해서 3장에서 정의한 수식

에 필요한 값을 설정하고 실험을 수행하였다.

본 논문의 실험은 Simgrid[5] 기반의 시뮬레이션을 통해 수행되었다. 성능 비교를 위해 사용되는 순차적 스케줄링은 스케줄러가 노드의 등록 순서대로 작업을 할당하는 반면, 제안 스케줄링은 <표 1>에 설정한 값을 기준으로 작업을 할당하였다.

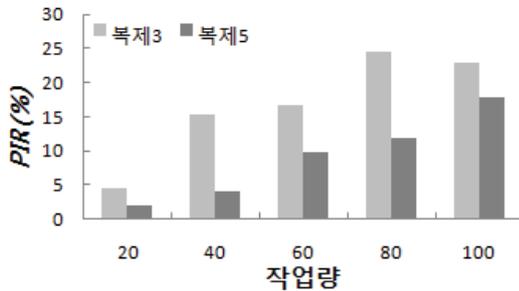
<표 1> 실험 파라미터

파라미터 정의	설정 값
수행 노드 결정 값 (T)	0.95
그룹의 수 (G)	3, 5, 7
작업 복제 수 (C_c)	3, 5

성능 비교는 제안한 작업 분배 스케줄러와 순차적 스케줄러를 이용하였다. 그리고 개별 스케줄러의 성능은 성능 향상 비율(PIR; performance improvement ratio)을 이용하여 측정하였다. PIR의 정의는 다음과 같다.

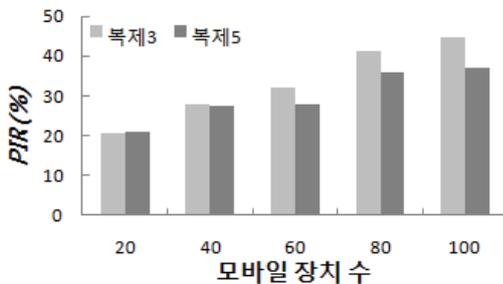
$$PIR(\%) = \frac{makespan_{serial} - makespan_{proposal}}{makespan_{serial}} \times 100 \quad (7)$$

(그림 5)에서는 작업량이 증가에 따른 성능 비교를 보여주며, (그림 6)에서는 모바일 수 증가에 따른 성능 비교를 보여준다. (그림 5)와 (그림 6)에서 작업 복제 개수로 3개와 5개를 사용하였다. (그림 5)에서 작업량 증가에 따른 PIR을 살펴보면, 복제 수가 3인 경우에는 평균 16.8%, 복제 수가 5인 경우에는 평균 9.1%의 성능 향상을 보였다.



(그림 5) 작업량에 따른 성능 비교

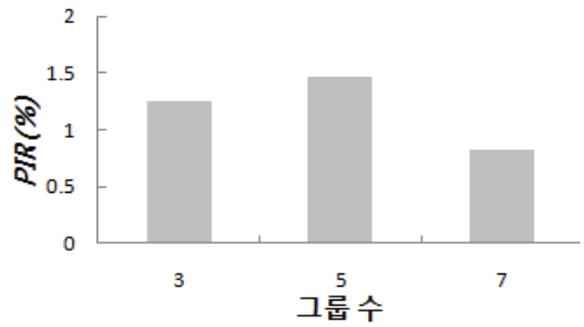
(그림 6)에서 모바일 장치 수의 증가에 따른 PIR을 살펴보면, 복제 수가 3인 경우에는 평균 33.3%, 복제 수가 5인 경우에는 평균 29.8%의 성능 향상을 보였다.



(그림 6) 모바일 장치 수에 따른 성능 비교

(그림 7)에서 그룹의 수에 따른 작업처리속도를 살펴보면, 그룹을 생성하지 않는 순차적인 스케줄링 알고리즘에 비해 그룹을 생성하는 제안한 스케줄링 알고리즘이 더 나

은 성능을 보여준다. 그룹의 수가 5개인 경우, 가장 우수한 성능 향상을 보여주고 있다.



(그림 7) 그룹 수에 따른 성능 비교

위와 같은 실험 결과들을 살펴볼 때, 신뢰성을 고려한 본 논문의 제안 기법이 신뢰성을 고려하지 않은 기존 기법에 비해 더 좋은 성능을 가진다는 것을 알 수 있다.

6. 결론

이 논문에서는 모바일 그리드 환경에서의 모바일 장치의 문제점을 해결하기 위해 모바일 장치의 신뢰성 기반의 작업 분배 및 작업 복제 기법을 제시하였다. 이 기법을 통해 신뢰성을 고려한 작업 분배가 이루어지고 모바일 장치의 결함 발생 문제를 해결하기 위해 여러 모바일 장치에게 작업은 복제하는 기법을 제시하였다. 성능 평가를 통하여 본 논문에서 제시한 스케줄링 기법이 작업 수행에 있어 성능 향상과 결함 방지 효과를 향상 시켰음을 증명하고 있다.

참고문헌

- [1] I. Foster, C. Kesselman, and S. Tueke "The anatomy of the grid: enabling scalable virtual organizations," Int. J. of High Performance Computing Applications, Vol. 15, No. 3, pp. 200 - 222, 2001.
- [2] G. Forman and J. Zahorjan "The challenges of mobile computing," IEEE Computer, Vol. 27, No. 4, pp. 38 - 47, 1994.
- [3] A. Litke, D. Halkos, K. Tserpes, D. Kyriazis, and T. Varvarigou "Fault tolerant and prioritized scheduling in OGSA-based mobile grids," Concurr. Comput. : Pract. Exper., Vol. 21, No. 4, pp. 533 - 556, 2009.
- [4] Y. Zhang, A. Mandal, C. Koelbel, and K. Cooper "Combined fault tolerance and scheduling techniques for workflow applications on computational grids," Proc. of 9th IEEE Int. Symp. on Cluster Computing and the Grid, pp. 244 - 251, 2009.
- [5] H. Casanova "Simgrid: A toolkit for the simulation of application scheduling." Proc. of 1st IEEE/ACM Int. Symp. on Cluster Computing and the Grid, pp. 430 - 437, 2001.