

Volunteer Computing 에서의 자원 상태전이 확률 테이블을 이용한 효율적인 작업 할당 기법

이재혁, 송충건, 박봉우, 유현창
분산클라우드컴퓨팅 연구실, 고려대학교
e-mail : {leedra, bongwoo, yuhc}@korea.ac.kr, security0730@naver.com

Efficient Task Allocation Technique Using Resources' State Transition Probability Table

Jaehyeck Lee, Chunggun Song, Bongwoo Lee, Heonchang Yu
Distributed & Cloud Computing Lab., Korea University

요 약

Volunteer Computing 은 대규모 프로젝트를 많은 분산된 데스크탑, 랩탑 PC, 그리고 스마트 폰과 같은 모바일 디바이스의 idle(유휴 상태)인 컴퓨팅 자원을 기여(volunteer)받아 수행하는 환경이다. 그러나 기여된 컴퓨팅 자원이 idle 과 used(사용 상태) 간의 상태전이가 빈번히 일어나거나, idle 로 지속되는 시간이 짧은 자원의 경우, 프로젝트를 수행하는데 비효율적이다. 따라서 본 논문에서는 이러한 자원을 구분하고, 해당 자원의 다음 상태전이 확률 테이블을 계산하여 결과에 따라 해당 자원에 적합한 작업을 할당하는 방법을 제안한다.

1. 서론

스마트폰, 태블릿 등 스마트 기기의 등장과 함께 많은 사람들이 사용하면서 많은 어플리케이션들이 존재한다. 특히 최근 몇 년 동안에는 페이스북과 같은 SNS 어플리케이션이 유행하면서 이러한 어플리케이션을 통해 각각의 사용자가 많은 데이터를 생성하고 있다. 이렇게 많은 데이터를 저장 및 관리하기 위해서 클라우드 서비스가 많이 사용되고, 의미 있는 데이터를 추출하고 결과를 분석하기 위해 클라우드와 같은 분산 컴퓨팅 기술이 사용되고 있다.

더불어 대부분의 사람들은 적어도 한 개 이상의 컴퓨터와 같은 컴퓨팅 디바이스를 갖고 있고, 컴퓨팅 성능도 향상되었다. 그러나 이러한 디바이스를 사용하지 않는 상태 (idle) 가 존재하는데, 이렇게 컴퓨팅 디바이스의 idle 상태인 자원을 네트워크로 연결하여 해결하기 힘든 난제나 컴퓨팅 자원이 많이 필요한 대규모 작업을 수행하는데 사용할 수 있도록 지원하는 Volunteer Computing 시스템이 사용되고 있다[1]. Volunteer Computing 의 대표적인 플랫폼으로 BOINC (Berkeley Open Infrastructure for Network Computing) 가 있다. Volunteer Computing 은 다음과 같은 장점을 갖는다[2].

- 연구나 프로젝트 등 컴퓨팅 자원이 부족한 곳에 충분한 컴퓨팅 자원을 공급할 수 있다.
- 컴퓨팅 자원 공급을 위한 비용을 절약할 수 있다.

Volunteer Computing 에서 자원을 공급하는 방법은

BOINC 와 같은 플랫폼을 통해 대규모 연구 또는 프로젝트를 등록하고, 자원을 기여 (volunteer)할 수 있다. 이때, 기여되는 자원은 idle 상태인 자원이다. 따라서 Volunteer Computing 의 프로젝트는 기여된 자원의 상태에 영향을 받는다. 예를 들면, 기여된 자원이 idle 상태에서 used 상태로 전이되면, 수행 중이던 작업은 중단된다. 또한, 기여된 자원이 idle 상태와 used 상태를 빈번하게 전이되는 경우, 작업을 수행하는 데에 비효율적이다. 따라서 본 논문에서는 이와 같은 문제를 위해 기여된 자원의 상태를 구분하여 자원에 대한 다음 상태전이 확률을 계산하여 각 자원의 상태 확률을 고려하여 적절한 작업을 할당함으로써 작업의 효율성을 높이기 위한 방법을 제안한다.

2. BOINC

BOINC 는 Volunteer Computing 을 지원하는 오픈소스 소프트웨어 플랫폼이다[3]. 처음에는 질병치료, 지구 온난화, 그리고 중성자별 발견 연구 등 컴퓨팅 자원이 많이 요구되는 과학연구를 위해 사용되었다. 대표적으로 SETI@HOME, SAT@HOME, 그리고 World Community Grid 프로젝트가 있으며, 과학연구를 위한 프로젝트뿐만 아니라 인공지능, 게임과 같은 소프트웨어 분야의 대규모 프로젝트도 존재한다. 그 이유는 IoT, 자율주행자동차 등 규모가 큰 연구가 진행되면서 그에 따른 프로젝트들이 존재하고 많은 컴퓨팅 자원이 필요하기 때문이다.

2.1 BOINC 의 구조

BOINC 플랫폼은 BOINC Server, BOINC Client, 그

리고 BOINC Manager 로 구성된다[4].

- BOINC Server 는 프로젝트들을 관리하고, 컴퓨팅 자원을 기여할 수 있도록 지원하는 Client Applications 을 제공한다. 또한, 프로젝트의 작업들을 여러 개의 작은 서브작업들로 나누고, 그것을 Client 에게 전송한다. 이후 Client 로부터 작업 처리 결과를 받는다.
- BOINC Client 는 서버로부터 작업을 받아 처리한다. 이후, 작업의 결과를 서버로 전송한다. 작업을 처리하기 위해 오직 idle 상태의 자원만 사용하고, 이것은 BOINC Manager 에 의해 관리된다.
- BOINC Manager 는 Client 의 자원을 관리하는 요소로, Client 가 자신의 자원을 얼마큼 기여할지, 또는 지정된 시간에만 작업을 처리할 수 있게 설정할 수 있게 도와준다. 또한 기여된 자원의 상태를 모니터링한다.

2.2 BOINC 의 한계점

BOINC Manager 는 기여된 자원에 대한 설정을 Client 에게 지원하고 자원의 상태를 모니터링하여 자원의 상태가 idle 일 때, 작업을 수행하고 used 상태일 때 작업 수행을 중지하는 방법으로 동작한다. 이러한 방법은 현재의 자원 상태만을 고려하기 때문에 작업 처리에 비효율적인 상황이 존재한다. 예를 들면, 기여된 자원의 상태가 idle 과 used 로 빈번히 전이되는 경우, 이에 따라 작업 수행과 작업 중지가 빈번히 발생하여 작업 처리에 비효율적이다. 따라서 본 논문에서는 BOINC 에서의 효율적인 작업 처리를 위해 자원 상태전이 확률을 계산하여 이를 기반으로 가장 적절한 작업을 할당하는 방법을 제안한다.

3. 제안하는 작업할당 기법

우선 컴퓨팅 자원의 상태는 idle 과 used 로 구분하고, 자원의 상태를 모니터링하는 주기를 $T (T \geq 0)$, 그리고 자원의 상태가 idle 에서 used 로 전이될 확률을 $x (0 \leq x \leq 1)$, used 에서 idle 로 전이될 확률을 $y (0 \leq y \leq 1)$ 라고 정의한다.

제안하는 기법은 상태전이확률테이블을 통해 자원의 다음 상태가 idle 또는 used 가 될 확률을 계산하여 그 결과를 기반으로 작업을 할당하는 것이다. 즉 마르코프 체인 모델[5]을 통해 $T + 1$ 시점의 자원 상태가 idle 과 used 가 될 확률 x, y 를 계산하여 이를 토대로 (표 1) 형태의 자원 상태전이확률테이블을 생성한다. 이 테이블에서 행과 열이 만나는 각 영역의 의미는 예를 들면, 행 idle 과 열 idle 의 교차점은 idle 상태에서 idle 상태로 전이될 확률을 의미한다.

(표 1) 컴퓨팅 자원의 상태전이확률테이블

상태	idle	used
idle	$1 - x$	x
used	y	$1 - y$

자원의 다음 상태전이 확률은 자원의 초기 상태와 상태전이확률테이블의 행렬 곱을 통해 계산된다. 예

를 들면, 자원의 초기 상태가 idle 이 될 확률이 $P_T(idle) = a (0 \leq a \leq 1)$ 이고 상태전이확률테이블이 (표 1)과 같을 때, 다음 상태전이 확률 P_{T+1} 은 다음과 같다.

$$P_{T+1}(idle, used) = (F_a(x, y), F'_a(x, y)) \quad (1)$$

$$F_a(x, y) = a(1 - x) + y(1 - a) \quad (2)$$

$$F'_a(x, y) = ax + (1 - a)(1 - y) \quad (3)$$

(그림 1)은 제안하는 기법의 전체적인 알고리즘을 나타낸다.

(그림 1) 제안기법의 알고리즘

```

1. while(1) {
2.  check the resource's state whether it's idle or not.
3.  calculate the  $P_{T+1}(idle, used)$  by using  $F_a(x, y)$ 
   and  $F'_a(x, y)$ .
4.  if  $F_a(x, y)$  is greater than  $F'_a(x, y)$ , then
5.    allocate the resource to computational tasks.
6.  sleep the time interval  $T$ .
7. }
```

4. 결론 및 향후 연구

본 논문에서는 각 컴퓨팅 자원의 상태전이 확률을 계산하여 idle 상태로 지속될 확률이 높은 자원과 idle 상태와 used 상태가 빈번히 변하거나 used 상태로 지속될 확률이 높은 자원을 구분하여 각 자원에게 적절한 작업을 분배함으로써 작업 수행을 효율적으로 수행할 수 있는 방법을 제안하였다. 향후 연구에서는 본 제안기법을 Volunteer Computing 환경에 적용하여, 컴퓨팅 자원의 상태를 고려하지 않고 수행하는 경우와 작업 수행 시간의 효율성을 검증할 것이다.

Acknowledgement

이 논문은 2017 년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-00587,(기반 SW-창조씨앗 1 단계) 분산 클라우드 기반 유무선 컴퓨팅 시스템의 유허자원 공유 플랫폼 개발).

참고문헌

- [1] M. N. Durrani, J. A. Shamsi, "Volunteer computing: requirements, challenges, and solutions", Journal of Network and Computer Applications 39 (2014), p369-380.
- [2] A. P. Afanasiev, I. V. Bychkov, M. O. Manzyuk, M. A. Posypkin, A. A. Semenov, O. S. Zaikin, "Technology for Integrating Idle Computing Cluster Resources into Volunteer Computing Projects", 2015 The 5th International Workshop on Computer Science and Engineering.
- [3] <https://boinc.berkeley.edu>.
- [4] D. P. Anderson, "BOINC: A System for Public-resource Computing and Storage", 5th IEEE/ACM International Workshop on Grid Computing (2004), p4-10.
- [5] W.R. Gilks, S. Richardson, D. Spiegelhalter, "Markov chain Monte Carlo in practice" book, 1995.