자발적 컴퓨팅 환경에서 중복작업 분배를 위한 P2P 기반 지역적 매치메이킹 기법

(P2P-based Regional Matchmaking Techniques for Distributing Redundant-works in Volunteer Computing Environments)

천 은 영 † 김 미 경 †
(Eun-Young Cheon) (Mi-Kyoung Kim)

국 승 학 †† 김 현 수 †††
(Seung Hak Kuk) (Hyeon Soo Kim)

요 약 자발적 컴퓨팅은 인터넷을 통해 연결된 컴퓨터의 유휴 자원을 활용해 큰 규모의 애플리케이션의 연산을 처리하는 컴퓨팅 패러다임이다. 자발적 컴퓨팅 환경에서 사용하는 중복 작업 분배기법은 동적인 자원과 작업의 특성에 대해 고려하지 않기 때문에 작업 시간 지연과 지속적인 작업재분배 요청으로 인해 작업 수행의 비효율성의 문제를 일으킨다. 이에 본 논문에서는 이러한 문제를 해결하기 위해 작업과 자원의 특성을 고려하여 작업을 재분배 할 수 있는 P2P 기반 지역적 매치메이킹 기법을 제안한다. 제안하는 매치메이킹은 작업/자원에 대한 명세, 자원가용공간 모니터링 모듈, 매칭 요청/응답을 위한 프로토콜, 매치율 계산 모듈 등의 요소로 구성된다.

키워드 : 자발적 컴퓨팅, 매치메이킹, 중복작업분배

Abstract Volunteer computing is a computing paradigm in which the operations of a large-scale application are processed using idle resources of Internet-connected computers. The redundant-work distribution techniques used in the volunteer computing environment cause some problems like delay of the work completion time or inefficiency of the work execution due to continuous requests of the work redistribution because the techniques do not consider the characteristics of works and dynamic resources. To cope with such problems this paper suggests a P2P based regional matchmaking technique which can redistribute works in consideration of the characteristics of the works and the participant resources. The techniques consist of the profiles for works/resources, the module for monitoring the available resources spaces, the protocols for the matching request/reply, the module for match-ratio calculation, and so on.

Key words : Volunteer Computing, Matchmaking, Redundant-work Distribution

1. 서 론

자발적 컴퓨팅(Volunteer Computing)은 인터넷에 연결된 수많은 유휴 컴퓨팅 자원들의 자발적 참여를 통해 고성능의 컴퓨팅을 창출하는 병렬처리시스템이다[1]. 자발적 컴퓨팅 환경에서 결과에 대한 신뢰성 부족과 주어진 시간 내의 처리결과에 대한 보장의 어려움을 해결하고자 제시된 중복 작업 분배기법[2]은 동적인 자원과 작업의 특성에 대해 고려하지 않기 때문에 작업 시간 지연과 작업 수행의 비효율성의 문제를 해결하지 못한다. 이러한 문제로 인해 서버의 지속적인 작업 재분배가 이루어져야 하며, 이는 결국 성능 측면에서의 장애 요인이 된다. 본 논문에서는 웹 서비스(Web Services)나 분산 자원관리시스템 등에서 공급/수요자간의 적합한 할당을 위해 사용되고 있는 매치메이킹 기법을 P2P기반 자발적 컴퓨팅환경에서 중복작업 분배기법에 도입하여 가장 적합한 작업이 자원제공자에 할당되어 중복작업 재분배율을 감소시키고, 전체작업완료시간을 단축하기 위한 지역적 매치메이킹 기법을 제안한다.

2. 관련연구

2.1 자발적 컴퓨팅 환경

자발적 컴퓨팅은 인터넷에 연결된 수많은 유휴 컴퓨팅 자원들의 자발적 참여를 통해 대량의 연산을 병렬처리 하는 방식으로 다음의 요소로 구성된다.

(1) 클라이언트 : 분산 애플리케이션 제공자로 대용량

· 이 논문은 2009 한국컴퓨터종합학술대회에서 '자발적 컴퓨터 환경에서 중복작업 분배를 위한 P2P 기반 지역적 매치메이킹 기법'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 충남대학교 전기정보통신공학부 컴퓨터전공
eycheon@cnu.ac.kr
lpgrane@nate.com
†† 정 회 원 : 충남대학교 전기정보통신공학부 컴퓨터전공
triple888@cnu.ac.kr
††† 종신회원 : 충남대학교 전기정보통신공학부 컴퓨터전공 교수
hskim401@cnu.ac.kr
(Corresponding author임)

논문접수 : 2009년 8월 14일
심사완료 : 2009년 10월 1일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제11호(2009.11)

분산처리가 필요한 자신의 분산 애플리케이션을 서버로 업로드하고 결과를 서버로부터 제공받는다[3].

(2) 자원 제공자(volunteer) : 자신의 컴퓨팅 자원을 통해 서버에서 전송한 분산 애플리케이션 작업을 수행하고 그 결과를 다시 서버에게 제출한다.

(3) 중앙관리서버 : 자원 제공자들을 통제하는 관리자로서 작업을 할당하고 스케줄링 한다.

자발적 컴퓨팅 시스템의 자원제공자는 불특정 이기종 PC들로 이루어져 작업완료시간에 대한 예측이 어려우며, 불안정한 컴퓨팅 환경과 네트워크단절, 보고되지 못한 이탈 등으로 인하여 분배된 작업의 결과를 돌려받지 못하는 등 전체 연산의 완료를 보장할 수 없거나, 결과수행시간이 오래 걸려 전체연산 성능시간을 지연시킬 수 있다.

2.2 매치메이킹(MatchMaking)

매치메이킹은 네트워크에 연결된 자원을 활용하기 위해 중앙의 매치메이커가 자원과 작업에 대한 명세를 관리하고, 그 명세를 기반으로 가장 적절한 작업과 자원의 연결을 위해 연결정보를 전달하는 방식이다[4,5].

2.3 Korea@Home 프로젝트

Korea@Home 프로젝트는 인터넷기반 분산컴퓨팅환경 구축 사업으로 자발적 컴퓨팅의 한 사례이다[2]. 이 환경에서는 메인서버와 통신한 에이전트가 대용량 애플리케이션 서버로부터 작업을 분배 받고 수행한 후 결과를 반환한다. 분배 된 작업 수행이 모두 완료된 에이전트는 에이전트간 통신을 통하여 중복작업을 분배받아 수행한다.

3. P2P 기반 중복작업분배 매치메이킹 기법

P2P기반 중복작업 분배기법은 서버의 부하를 줄이고 작업결과의 신뢰성을 확보하고 서버 붕괴 시에도 연산 지속력 및 결함 포용력을 높일 수 있다[4]. 그러나 중복작업 분배 시 수행 작업 및 환경의 특성들이 고려되지 않음으로 인하여, 작업수행 중 작업만료시간이 지연되거나 서버로부터 작업을 재분배야 하는 문제가 발생하는 등 작업과 자원의 매치불균형으로 시스템 성능저하가 발생하게 된다.

3.1 매치메이킹 기법

본 논문에서 제안하는 매치메이킹 알고리즘은 자원제공자의 동적 가용성을 고려하여 시간적 가용성과 공간적 가용성으로 구분한다. 시간가용성은 자원 사용의 시간의 양으로 다양한 연속적 시간함수로 표현되는 가용성을 의미하는데, 자발적 컴퓨팅에 참여한 자원제공자가 실제 작업에 참여한 시간을 이용해 자원제공자의 시간 가용성을 판단하는 지표로 사용한다. 공간적 가용성은 자원의 물리적 용량(Physical capacity)과 공간적 능력을 측정하는 단위로 표현되는 가용성을 의미하며, 본 논문에서는 작업의 수행 특성으로 가장 두드러진 CPU가용성과 메모리 가용성의 동적변화를 고려한다. 매치율 계산 시 위의 시·공간적 가용성들을 자원의 가용성으로 활용하여 작업과의 적합성을 판단한다.

3.1.1 동적 가용성기반 매치메이킹

Condor[6]나 Web Services에서 보듯 안정된 환경에서 명세를 통한 매치메이킹 기법이 자발적 컴퓨팅환경에 적용되기 위해서는 자원휘발성이 존재하는 환경에 대한 고려가 필요하다. 매치메이킹 기법은 자원이 시스템에 참여하면서 자원의 정적 정보를 매치메이커에게 광고하고, 매치메이커는 이 정보를 바탕으로 하여 가장 적절한 작업을 매칭하게 된다. 분산컴퓨터 환경에서 작업 완료시간은 연산에 참여한 개별 자원제공자의 성능과 자원제공자의 가용성에 의존하는데, 자원제공자의 자율성과 휘발성으로 인해 안정적 작업 수행이 보장되지 않는다. 사용자의 유휴자원만을 사용하는 자발적 컴퓨팅 환경에서 정적인 자원정보(메모리 또는 CPU 성능)만을 고려하는 것은 자원제공자의 동적 수행환경 변화에 의한 작업지연으로 서버로부터의 작업 재분배를 야기하여, 전체적인 작업의 완료시간을 지연시킬 수 있다. 따라서 작업분배 시 자원의 동적 가용성 정보를 활용한 분배알고리즘이 요구된다.

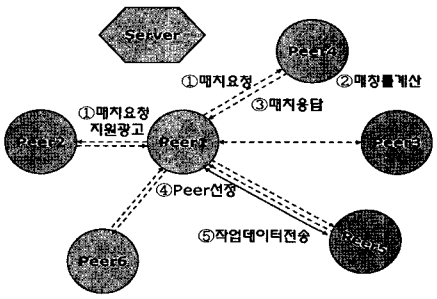

그림 1 동등계층 Peer간 지역적 매치메이킹

3.1.2 자원제공자 간 중복작업분배를 위한 지역적 매치메이킹

지역적 매치메이킹 방식은 P2P환경의 Peer들 중에서 수퍼Peer 역할을 하는 자원제공자가 매치메이커의 역할을 수행한다. 수퍼Peer의 선정 시 리프노드 개수 제한, 지역성 등 많은 고려 요소가 필요하며 자원의 안정성이 떨어지는 자발적 컴퓨팅 환경에서는 수퍼Peer의 선정을 통한 오버레이 네트워크를 구성하더라도 자원의 참여 이탈이 잦아 네트워크 재구성을 위한 오버헤드가 많아 적용하기에 적절치 않다. 따라서 본 논문에서 제안하는 매치메이킹 기법은 동등계층의 Peer들 간의 지역적 매치메이킹 방식으로 그림 1에서 보듯이 모든 자원제공자가 작업의 요청, 매치율 계산 및 응답, 적합한 작업 선

정의 기능을 동일하게 가지므로 Peer간의 유기적 통신을 통해 매치메이킹을 수행하는 특징이 있다.

3.2 제안 매치메이킹 구성요소

3.2.1 작업/자원 정보 명세

매치메이킹을 하기에 앞서 자원과 작업에 대한 명세가 선행되어야 한다.

(1) 작업명세 : 작업에 대한 자원 의존도에 따른 가중치에 대한 태그들(메모리 가중치, Flops 가중치, 평균 수행시간에 대한 가중치)을 작업명세서에 추가하였다(그림 2). 서버로부터 작업데이터와 함께 자원제공자에게 전송되는 작업명세는 관리자가 작업을 서버에 업로드할 때 작성되며, DB에 등록되어 관리된다.

(2) 자원제공자의 자원정보 명세 : 자원제공자의 자원정보 명세에는 매치율 계산 시 활용될 작업 수행시점에 예상되어지는 시·공간적 가용성을 표현한다(그림 3).

```
<joblist>
  <appid></appid>
  <projectid></projectid>
  <policy>
   <platform></platform>
   <max_runtime></max_runtime>
   <os></os>
   <mem></mem>
   <lib></lib>
   <hdd></hdd>
   <RATE_mem></RATE_mem>
   <RATE_flops></RATE_flops>
   <RATE_time></RATE_time>
  </policy>
  <commonfiles>..</commonfiles>
  <workid></workid>
</joblist>
```

그림 2 작업 명세

```
<cominfo>
 <cpu num="" cat="" clock=""></cpu>
 <memory physical=""></memory>
 <os cat="" ver=""></os>
 <flops></flops>
 <A_mem></A_mem>
 <A_flops></A_flops>
 <avg_times></avg_times>
</cominfo>
```

그림 3 자원 명세

3.2.2 자원 동적 가용성 예측모듈

자원의 동적 가용성을 기반으로 하는 매치메이킹 수행을 위해, 자원 시스템 가용성의 최근 값에 더 가중치를 두어 예상되는 가용율을 예측/판단하고, 매치율 계산 시 자원 동적 요소로 사용한다.

(1) 자원 예상 가용공간 모니터링 : 자원(메모리, CPU)의 가용성을 고려하는 매치메이킹은 사용자의 이전 이용 패턴을 바탕으로 시간주기를 정해 현재 자원의 가용성을 모니터링 하고 그 값을 기준으로 하여 예상가용공간을 예측한다.

① T_n : n번째 자원 모니터링 주기

② A_n : T_n 자원가용 공간

③ PA_n : T_n시 예상 자원가용 공간

④ 시간 T_n의 예상 가용 공간 계산

$$PA_n = Avg(PA_{n-1} + A_n) = \frac{(PA_{n-1} + A_n)}{2} = \frac{A_0 + 2^0 A_1 + \cdots 2^{n-1} A_n}{2^n}$$

시간이 거듭될수록 가장 초기 가용 공간 A_0의 예상 가용공간을 구하는데 미치는 영향은 $A_0/2^n$로 매우 미미해진다. 그러나 가장 최근 모니터링된 가용 공간 An은 $(2^{n-1}/2^n)A_n$의 가중치로 적용되며 자원 예상 가용 정보로 활용한다.

3.2.3 매치율 계산

작업 명세서에 들어있는 작업의 수행특성에 따른 자원의존도 요율과 자원 가용성을 기반으로 자원과 작업간의 매치율을 판단하는 최종 계산방법은 다음과 같다.

(1) 중복 작업 분배 매치율 알고리즘: 중복작업분배는 서버로부터 분배받은 애플리케이션의 작업을 수행하고 나면, 자신이 인근의 자원제공자에게 중복작업분배를 요청하는 방식으로 이루어진다.

① 정적정보를 기반으로 최소 요구사항 확인

② 동적정보를 기반으로 작업/자원 매치율 계산

③ 매치율 정렬을 통한 작업 선정

(2) 정적정보를 기반으로 최소요구사항 확인 : 작업을 요청하는 자원제공자의 기본사양이 작업의 최소제약조건을 만족하는가를 판단한다.

(3) 가용성기반 작업/자원 매치율 계산 : 작업과 자원제공자 자원의 시·공간적 가용성을 기반으로 매치율을 계산한다.

$$Matching_{ratio} = SMem_{pos}(Mem_{pspace}(peer_x))* Mem_{appsfactor}(apps_y) + SFlops_{pos}(Flops_{pspace}(peer_x))* Flops_{appsfactor}(apps_y) + Time_{avail}(peer_x)* Time_{appsfactor}(apps_y)$$

- $Mem_{pspace}(peer_x)$: 현재를 기준으로 Peer가 이후 작업이 수행될 때 예상되는 메모리 가용공간
- $Flops_{pspace}(peer_x)$: 예상가용 CPU성능(Flops)
- $SMem_{pos}()$, $SFlops_{pos}()$: 전체 자발적 컴퓨팅 환경에서의 메모리 자원과 CPU 성능을 0~Max(자원공간)까지의 값을 균등분할 했을 때 해당 예상 가용공간의 시스템내의 자원의 위치를 나타낸다. 이것은 매칭계산 시 메모리와 CPU의 기본 단위가 다르므로 그 기준을 동일하게 해주는 동시에 시스템내의 유용성을 백분율로 표현하도록 한다. 이는 시스템관점에서 자원이 갖는 가치를 의미한다.
- $Mem_{appsfactor}(apps_y)$, $Flops_{appsfactor}(apps_y)$: 작업의 자원 의존도에 대한 가중치

3.3 제안 매치메이킹 시나리오

제안하는 매치메이킹의 흐름은 다음과 같다(그림 4).

(1) 수행 작업을 소진한 자원제공자가 주변의 자원제공자에게 자신의 동적예측가용성정보와 함께 중복작업 매칭을 요청한다.

(2) 매칭요청 시 받은 동적예측가용성 정보와 작업 큐에 존재하는 작업 명세에 정의된 자원의존도 요율로 매치율을 구한다.

(3) 매치율을 요청했던 자원제공자에게 전송한다.

(4) 요청했던 자원제공자는 전송받은 매치율 중 가장 높은 작업을 선택한다.

(5) 선택된 자원제공자로부터 중복작업을 받아 수행한다.

이 흐름을 작업 요청자와 작업 응답자의 관계로 보면 그림 5와 같다.

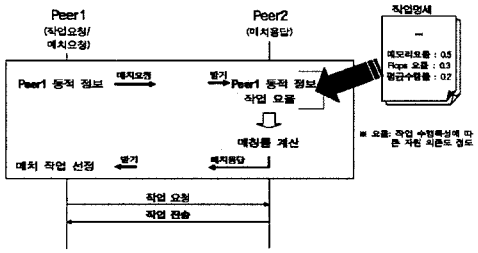

그림 4 작업 요청지/작업 응답자 간의 유기적 매치메이킹 흐름

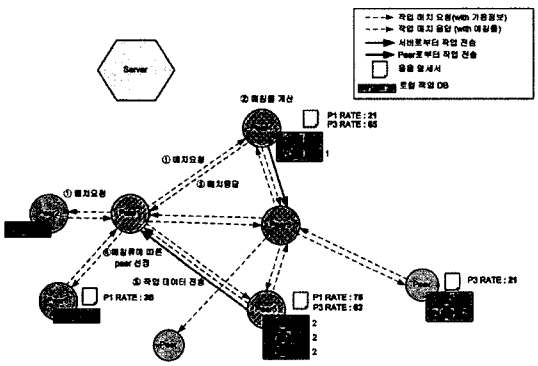

그림 5 중복작업분배를 위한 지역적 매치메이킹 시나리오

3.4 시스템 설계

본 논문에서 제안하는 작업의 흐름도는 그림 6, 그림 7과 같으며, 자원의 제공자는 아래의 모든 기능들을 포함해야 한다.

(1) 매치요청/응답대기

(2) 작업 정적 요구사항확인 / 매치율 계산

(3) 매치 응답 처리

다른 Peer로부터 매치 요청을 수신하였을 때, 작업 큐에 작업의 중복수가 1보다 크면 해당 작업이 요청한 자원제공자의 환경(정적환경)에서 수행될 수 있는지 애플리케이션의 수행요구사항 만족여부를 살핀다. 이것을 만족하면 Peer와 애플리케이션간의 매치율을 구해, 매치 응답 메시지를 요청 자원제공자에게 전달한다(그림 7).

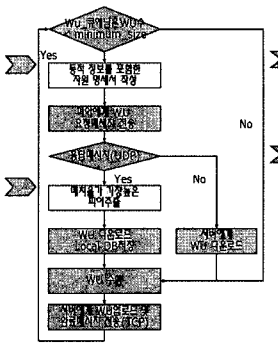

그림 6 매칭 요청 및 작업 요청 흐름도

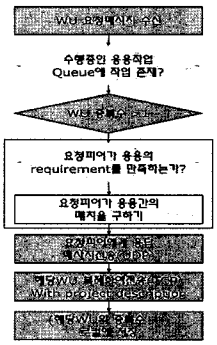

그림 7 매치 응답 및 작업 분배 흐름도

3.5 시스템 구현

본 논문에서 제안한 기법을 활용하기 위하여 Korea@Home의 시스템 구조에 매치메이킹 모듈과 동적정보 모니터링 모듈을 추가하였다(그림 8). 자원제공자(에이전트)는 서버로부터 작업을 수행하는 코어 모듈과 애플리케이션 수행 모듈로 크게 나뉘며, 코어 모듈은 애플리케이션 수행을 위한 작업관리와 P2P 통신모듈로 이루어져있다.

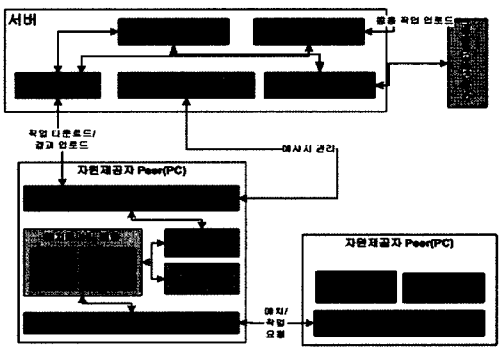

그림 8 매치메이킹 모듈을 포함한 전체 시스템 구조도

4. 성능평가

가상탐색(Virtual Screening) 기술을 이용하여 약이 체내에서 특정 작용점(수용체, 효소 등)과 결합해 약효가 발휘되는 것을 PC로 시뮬레이션 하는 신약후보물질 탐색과 RSA 암호 방식의 암호 키를 생성하기 위한 소수를 찾는 인수분해기반 암호분석의 애플리케이션을 Korea@Home에서 동시에 수행하였다.

실험 비교 대상은 동일한 개수의 애플리케이션을 수행하는 P2P기반 중복작업 분배기법과 본 논문에서 제안하는 P2P기반 지역적 매치메이킹 기법이다.

4.1 작업 전체 수행시간 비교

그림 9는 인수분해분석과 신약후보물질탐색의 전체 수행시간을 나타낸다. 그래프에서 보는 바와 같이 중복작업분배기법(RWD)에 비하여 본 논문의 방법인 P2P기반 매치메이킹(RWD_M)이 약 10%의 향상이 있었다.

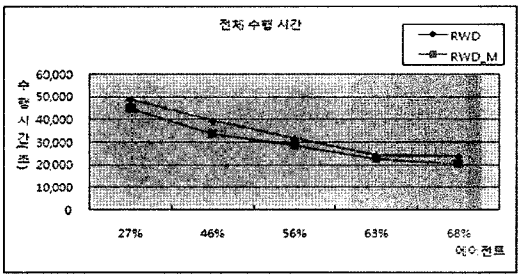

그림 9 전체 수행시간 비교

4.2 작업 재분배로 인한 시간 비교

작업 셋을 수행하면서 모든 애플리케이션을 수행할 수 있는 PC 7대와 인수분해가 수행될 수 있는 PC를 3, 6, 9, 12, 15개씩 증가하여 테스트를 수행하였을 때 작업결과 시간의 지연이나 기타 요인으로 인하여 서버로부터 재분배 한 결과이다. 신약후보물질탐색의 경우 재분배시간지연 측면에서 RWD_M이 RWD에 비해 약 22%의 향상을 보였다(그림 10).

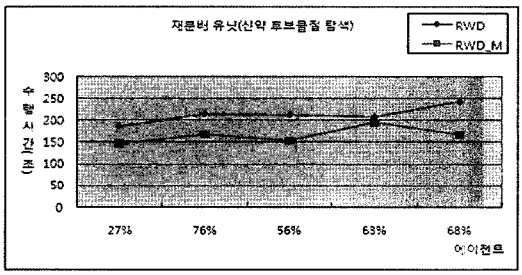

그림 10 작업 재분배로 인한 시간 비교 (신약후보물질 탐색)

그림 11에서 보듯이 인수분해 탐색의 경우 재분배로 인한 시간 지연 측면에서는 RWD_M이 RWD에 비해 약 40%의 향상을 보였다. 이는 함께 수행한 신약 후보물질 탐색보다 월등히 향상된 것으로써, RWD가 매치메이킹 없이 작업을 분배할 뿐 아니라 인수분해의 자원특성상 메모리에 영향을 많이 받는데 동적상황에 대한 고려 없이 분배함으로 인하여 작업의 완료시간이 지연된 경우에 서버로부터 작업을 재분배 받았기 때문이다.

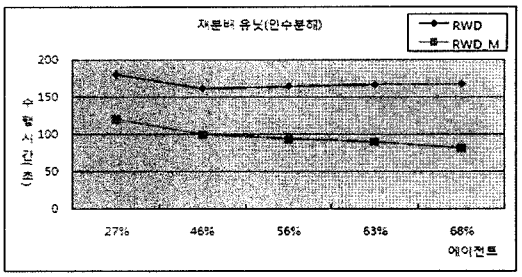

그림 11 작업 재분배로 인한 시간 비교(인수분해)

5. 결론 및 향후연구과제

본 논문에서는 자발적 컴퓨팅 환경에서 작업수행 특성에 따른 자원의존도를 명시하고 그 정보를 고려하여 작업에 대한 자원의 적합성을 고려한 매칭을 통해 가장 적절한 자원을 선정하여 수행시간을 최소화하여 서버의 재분배율을 줄이고자 하였다. 실험을 통해 본 논문에서 제안한 작업 분배 방식이 재분배율과 전체 수행시간을 감소시킨다는 것을 알게 되었다.

그러나 동등 Peer간의 지역적 매치메이킹을 위해 많은 메시지들이 오가므로 통신비용으로 인한 오버헤드가 발생할 수 있고, 애플리케이션의 특성 파악을 잘못한 경우 오히려 분배가 잘 되지 않는 잘못된 결과를 낼 수 있으므로 애플리케이션의 특성을 파악하고 의존도를 정량화 하는 방안에 대한 연구가 필요하다.

참 고 문 헌

[1] Luis, F.G.S. and H. Satoshi, Bayanihan: *building and studying web-based volunteer computing systems using Java*, Elsevier Science Publishers B. V., pp.675-686, 1999.

[2] J.H. Choi, et al., "Non-disturbance Scheduling based on User Usage Pattern for Volunteer Computing," *Proc. of Int'l Conf. on Convergence Technology and Information Convergence*, 2007.

[3] Anderson, D.P., E. Korpela, and R. Walton, "High-performance task distribution for volunteer computing," *Proc. of First Int'l Conf. on e-Science and Grid Computing*, 2005.

[4] Uppuluri, P., et al., "P2P grid: service oriented framework for distributed resource management," *Proc. of Int'l Conf. on Services Computing*, 2005.

[5] Chakravarti, A.J., G. Baumgartner, and M. Lauria, "The organic grid: self-organizing computation on a peer-to-peer network," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol.35, no.3, pp.373-384, 2005.

[6] Alain Roy, M.L., *Condor and preemptive resume scheduling*, pp.135-144, 2004.