

# CPU공유를 위한 계층적 P2P 모델

윤기철<sup>0</sup> 임대영 김길용  
부산대 컴퓨터 공학과

herofe@chollian.net dylin@hyowon.pusan.ac.kr gykim@hyowon.pusan.ac.kr

## Hierarchical P2P Model for CPU sharing

Ki-Chul Yun<sup>0</sup> Dae-Young Lim Gil-Yong Kim  
Pusan National University Computer Engineering

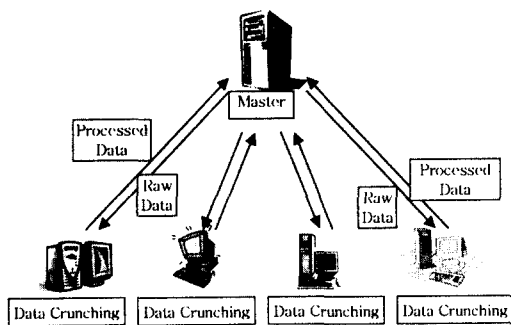
### 요 약

지금까지의 분산 컴퓨팅 시스템은 클라이언트의 요청을 서버에서 분산처리하는 방식이었으나, SETI는 피어투피어 기술을 이용하여 이와 반대로 서버의 요청을 클라이언트에서 분산처리하는 방식을 취하고 있다. 여기서 데이터의 요청과 분산처리가 모두 클라이언트에서 이루어지는 분산시스템을 구성해볼 수 있다. 본 논문에서는 피어투피어 모델에 분산컴퓨팅을 적용하여, 기존 모델의 문제점을 해결한 새로운 피어투피어 모델을 제시해보고자 한다.

### 1. 서 론

지금까지 연구되어왔던 분산시스템은 모두 서버에 의한 데이터처리에 초점이 맞추어져왔다. SETI는 외계전파를 분석하는데 드는 엄청난 컴퓨팅 파워를 얻기 위해 이제까지의 고비용의 서버를 이용한 분산컴퓨팅 시스템을 구성하는 대신에, SETI@home[1]이라는 프로젝트를 통해 클라이언트들의 컴퓨팅 파워를 이용하였다.

SETI@home은 지금까지의 분산컴퓨팅 시스템과는 반대로 서버가 클라이언트에게 데이터처리를 요청하고 클라이언트에 의해 처리된다는 것이다. SETI@home의 데이터 흐름은 <그림 1>과 같다.



<그림 1> SETI@home의 데이터 흐름도

SETI@home에서 사용한 피어투피어 기술은 전혀 새로운 기술이 아니다. 이미 과거부터 사용되어 왔던 기술이며 이 기술은 파일 공유, 리소스 공유, 메신저, 협동시스템[2] 등에

널리 사용되고 있다. 피어투피어에서의 피어의 개념은 대부분의 거대 네트워크의 구성인 클라이언트/서버 모델에 대응되는 개념으로 누가 클라이언트이고 누가 서버인지를 구별짓지 않고, 네트워크에 참여한 구성원이 모두 동등한 지위를 갖는 것이다[3]. 그러나 SETI@home의 피어들은 자신의 컴퓨팅 파워를 제공하는 수동적인 역할만을 수행하고 있다. 여기서 피어가 컴퓨팅파워를 요구하고 이를 제공하는 진정한 의미의 분산컴퓨팅(CPU 공유)시스템을 생각해 볼 수 있다. 피어투피어 시스템은 피어의 정보(메타데이터)를 어떻게 관리하느냐에 따라 하이브리드 피어투피어, 퓨어 피어투피어로 나뉘어 진다. 하지만 이 모델은 각각 서버의 용량에 의해 참여할수 있는 피어들의 수가 정해져 있다는 것과 스램의 발생이라는 단점을 가지고 있다. 이를 해결하기 위해 계층적인 형태를 가지는 새로운 피어투피어 모델을 제시해보고자 한다.

### 2. 관련 연구

CPU를 공유하기 위해서는 퍼스털 컴퓨터의 사양과 제공가능한 컴퓨팅 파워, 피어의 네트워크의 속도등의 메타데이터를 관리하여야 한다.

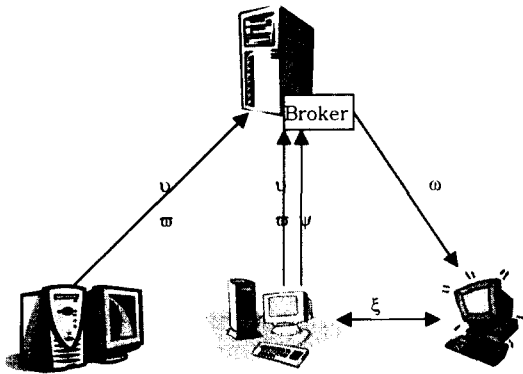
이 메타데이터를 어떻게 관리하느냐에 따라 하이브리드 피어투피어와 퓨어 피어투피어로 나누어진다.

#### 2.1 하이브리드 피어투피어

하이브리드 피어투피어는 순수한 피어투피어 방식이 아니다. 냅스터로 대표되는 이 모델은 메타데이터를 처리하는 서버(브로커)를 따로 두고 있다. 피어는 이에

접속후 자신의 메타데이터를 브로커에 등록하고 원하는 메타데이터를 브로커에 요청한다. 브로커는 알맞은 피어를 검색하여 다시 요청한 피어에 보내며, 피어는 브로커가 제공하는 정보를 바탕으로 해당피어를 찾고 서비스를 요청하는 방식이다.

이 모델을 CPU공유에 적용하면 <그림2>와 같다. <그림2>에서는 하나의 피어에게만 서비스 요청하는 것처럼 보이나, 실제로는 피어들의 리스트를 브로커에게서 받아서 피어들에게 분산처리를 요청하게 된다.



- α 중앙 서버로 자신이 구동인 것을 알린다.
- β 피어의 정보를 서버에 등록한다.
- γ 서비스가능한 피어를 검색한다.
- δ 피어에게 처리할 데이터를 요청하고 처리 결과를 받는다.
- ψ 변환된 피어의 정보를 서버에 수정 요청한다.

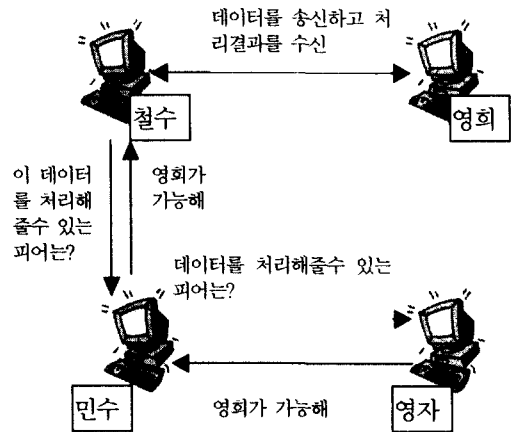
<그림 2> 하이브리드 피어투피어의 데이터 흐름

일반적인 파일공유시스템과는 달리 CPU공유를 할 경우 공유가능한 컴퓨팅 파워가 변하므로 서버는 이를 실시간으로 업데이트를 해야 한다. 서버의 기능을 최소화하였다고는 하나 접속한 피어의 수가 많아지고 처리요청이 많아질수록 서버에 부하가 거질 것이므로 이 모델은 서버의 용량에 의해 네트워크에 참여할 수 있는 피어의 수는 한정된다.

### 2.2 피어 피어투피어

따로 서버가 필요없이 모두 피어들로만 구성되는(따로 서버가 필요없는) 피어투피어 모델이다. 이 방식은 그누텔라 [4]에 의해 채택되고 있으며 많은 피어투피어 프로그램이 이 모델로 구현되어 있다.

각 피어들은 한정된 이웃 피어들의 메타데이터를 관리한다. 각 피어들은 서로에게 필요한 정보를 요청하며 피어들중 해당사항이 있으면 이는 요청한 피어에게 보내지게 된다. 이 방식은 스몰월드이펙트[5]에 의해 설명되어 질 수 있다. 즉 자신이 원하는 메타데이터를 어느 피어가 가지고 있을 확률이 높으며 그 피어와는 패스는 받아들일 수 있는 거리에 존재한다는 것이다



<그림 3> 피어 피어투피어의 데이터 흐름

이와 같은 피어 피어투피어 모델은 서버가 필요없으며 네트워크가 자생적이며 동적이라는 것이 장점을 가진다. 그러나 피어 피어투피어의 단점은 스팸의 발생한다는 것이다. 예를 들어 네이버 슬랏이 10이고 TTL이 5인 쿼리패킷을 보낸다면 이 패킷은 10의 5승인 10만개의 패킷으로 늘어나게 된다.

### 3. 계층적 피어투피어 모델

이상에서 알아본 바와 같이 두 가지 모델은 접속가능한 피어들의 수가 한정된다는 것과 스팸의 발생이라는 단점을 가지고 있다. 이러한 단점을 해결하기 위한 방법으로써 본 논문에서는 계층적 피어투피어 모델(hierarchical peer-to-peer model)을 제시코자 한다.

계층적 피어투피어 모델의 유사한 형태로는 DNS를 들 수 있다. 하지만 DNS와 같은 정적인 모델과는 달리 피어투피어 모델은 매우 동적이다. 따라서 자체적으로 계층을 구성하도록 설계할 필요성이 있다. 이 네트워크는 피어그룹이라는 단위로 구성되며 각 피어그룹은 코어디네이터에 의해 관리되어 진다.

#### 3.1 피어그룹

코어디네이터에 의해 관리되어지는 피어들의 모임이다. 피어그룹은 그 자체가 하이브리드 피어투피어 모델의 축소판이라고 볼 수 있다. 이 피어그룹은 전체 구성원으로써 최소 단위이자 최대 단위라 할 수 있다. 그리고 피어그룹의 크기는 코어디네이터의 메타데이터를 저장하는 슬랏에 의해 최대 피어의 수가 정해진다. 피어그룹은 레벨이 있으며 상위 레벨의 피어그룹에서의 피어는 하위 레벨의 피어그룹과 코어디네이터를 의미하게 된다. 하위레벨의 피어그룹은 상위레벨에서는 하나의 피어로 다루어지며 이 때의 메타 데이터는 하

위레벨의 코어디네이터의 피어의 정보와 하위레벨의 그룹 전체의 가용량이다.

### 3.2 코어디네이터

하이브리드 피어투피어에서의 브로커의 역할을 수행하는 피어이다. 코어디네이터는 피어그룹내의 모든 피어들(코어디네이터 포함)의 메타데이터를 관리하며 그룹내의 가용량이 요청량을 넘어설 경우 상위 레벨의 코어디네이터에게 부족분에 대한 처리 가능한 피어의 검색을 요청한다.

### 3.3 알고리즘

#### 3.3.1 피어그룹에의 접속

```

Join(level)
{
    if NoPeerGroup then
        CreatePeerGroup();
        BeCoordinator();
    elseif PeerGroupJoin() fail then
        level = level+ 1;
        Join(level);
    end if
}
    
```

#### 3.3.2 피어그룹에 요청

```

Request(requested_CPU)
{
    if CPUofThisGroup > requested_CPU then
        Allocate();
        If level = 1 then
            Return ListOfPeer();
        Else
            RequestListOfPeer(level-1);
        End if
    Else
        Request(requested_CPU-CPUofThisGroup);
    End if
}
    
```

이러한 방식으로 구성된 피어그룹은 바텀업 형식의 트리 구조를 띄게 되며 퓨어피어투피어와 같이 순수한 피어들에 의해 이루어진다.

### 4. 결론 및 향후 계획

본 논문에서 제시된 피어투피어모델은 하이브리드 피어투피어 모델을 계층적으로 구성한 방식으로써 퓨어피어투피어에서 발생하는 스팸 문제를 해결하였으며, 코어디네이터의 피어의 한계치를 넘을 때 새로운 계층을 동적으로 생성함으로써 접속 가능한 피어의 수가 제한되는 문제를 해결하였다.

향후 연구 과제로는 피어들이 접속함으로써 발생할 수 있는 피어들의 집중화 현상을 해결할 수 있는 알고리즘에 대한 연구이다.

### 5. 참고문헌

[1] <http://setiathome.ssl.berkeley.edu>  
 [2] *Fox, G. Computing in Science & Engineering , Volume:3 Issue:3, May-June 2001 Page(s): 75 -77*  
 [3] 이재규, 신세대 네트워크의 키워드 ‘P2P’의 힘 마이크로 소프트웨어 2000년 10월호  
 [4] <http://gnutella.wego.com>  
 [5] D.J. Watts and S.H. Strogatz, “Collective Dynamics of ‘Small-World’ Networkds,” Nature 393, p.440(1998)  
 [6] Andy Oram PEER-TO-PEER Harnessing the power of Distructive Technologies