論 文
55D-2-3

차별화된 성능 기반을 이용한 Hybrid P2P 시스템

An Intelligent Performance based Hybrid P2P System for Distributed Resource Sharing

閔 秀 泓† · 趙 東 燮*
(Suhong Min · Dongsub Cho)

Abstract - The emerging peer-to-peer (P2P) model has recently gained a significant attention due to its high potential of sharing various resources among networked users. Super-peer based unstructured P2P systems have been found very effective by dividing the peers into two layers, super-peer and ordinary-peer. Super-peers deal with all queries instead of ordinary-peers. The existing P2P systems assume all super-peers have equal responsibility and capabilities even if all super-peers have relative advantages and disadvantages. In this paper, we suggest the ISP2P (Intelligent Super-peer based P2P) which allows to select the best super peer to ordinary-peers. We classify super peers according as the capacity of an ordinary-peer to a super-peer. We show that these considerations could improve the performance of the response time and provide higher quality results to all peers in the network.

Key Words : P2P (Peer-to-Peer), Super-peer, Capacity, Hybrid P2P system

1. 장 서 론

인터넷 사용량의 증가와 네트워크 속도의 증가는 기존의 서버와 클라이언트의 수동적인 자원 공유에서 하나의 노드가 서버와 클라이언트의 역할을 동시에 하는 P2P 시스템으로 발전하였다. P2P 시스템의 이러한 특징은 피어들에게 분산되어 있는 자원 즉, 컴퓨팅 능력, 저장 공간, 컨텐츠 등을 자율적으로 공유할 수 있도록 하였다. 현재 P2P 시스템에 관한 다양한 연구가 진행 중이다 [1,2]. 그러나 현재 연구되어 오고 있는 대단수의 P2P 모델은 네트워크에 참여하는 피어들을 모두 동등한 능력을 갖고 있다고 가정한다. 그러나 이렇게 설계된 P2P 시스템에서는 특정 노드에 자원이 집중되어 병목현상(bottleneck) 이 야기 되거나 성능이 낮은 피어로 인해 전체 네트워크 속도가 크게 저하 될 수 있다. 따라서 이러한 문제점을 해결하기 위해서 본 논문에서는 네트워크 내에 있는 노드들을 성능에 따라서 차별화 하였다. 먼저, 수퍼 피어의 개념을 이용해서 성능에 따라 일반 피어와 수퍼 피어로 노드들을 분류 하였으며, 성능이 우수한 수퍼 피어는 일반 피어의 쿼리를 대신 처리해 주는 역할을 하도록 한다. 이러한 수퍼 피어의 개념은 과거 몇몇 연구에서도 사용되었다. 그러나 기존의 연구에서는 수퍼 피어로 분류된 피어들을 모두 동등하게 처리하였으며, 일반 피어는 랜덤하게 수퍼 피어를 선택해서 쿼리를 대신 처리하도록 하였다. 본 논문에서는 기존의 연구와 차별화되게 수퍼 피어와 일반 피어의 성능의 관계를 분석해서 일반 피어가 최적의 수퍼 피어를 선택할 수 있도록 한다. 일반 피어들은 쿼리를 처리하기 위해 수퍼 피어를 선택하며, 이 과정에서 일반 피어의 성능 대 수퍼 피어의 성능을 분석해서 최적의 수퍼 피어를 선택하도록 한다. 수퍼 피어의 성능은 크게 2가지 부분으로 나누어 분석한다. 먼저, 일반 피어와의 거리, CPU 능력, 쿼리 처리에 소요되는 응답 시간 등을 통해 일반 피어와의 성능을 측정하며, 둘째, 수퍼 피어의 쿼리 처리능력- 최대 결과물, 최대 성능, 최대 속도에 대해 평가한다. 본 논문에서는 수퍼 피어대 일반 피어의 성능을 바탕으로 최적의 수퍼 피어를 통해 일반 피어가 쿼리를 효율적으로 처리할 수 있도록 한다. 또한 적절한 수퍼 피어의 선택으로 대역폭(bandwidth), 응답 시간 등을 향상시키고자 한다.

† 교신저자, 正會員 : 이화여자대학교 컴퓨터학과 박사과정
E-mail : shimin@ewhain.net
* 正 會 員 : 이화여자대학교 컴퓨터학과 정교수
接受日字 : 2005年 11月 17日
最終完了 : 2005年 12月 14日

2. 장 관련 연구

2.1. 수퍼 피어의 정의

수퍼 피어는 기존의 일반 피어와 같은 역할, 즉 서버와 클라이언트 역할을 동시에 수행하며 또한 일반 피어와는 달리 수퍼 피어에 연결된 피어들의 쿼리를 대신 처리해 주는 역할을 한다. 기존의 순수 P2P 시스템에서는 쿼리를 받은 피어는 이를 각자가 전달해주는 반면, 수퍼 피어를 사용하는 시스템에서는 일반 피어들은 자원을 요청할 경우, 수퍼 피어에게 쿼리를 전송하며, 수퍼 피어는 수신한 쿼리를 어디로 전송할지를 결정한다. 또한, 수퍼 피어는 요청한 쿼리에 대한 결과에 대해 응답 메시지를 생성하여, 요청한 피어에게 전송해준다. 이러한 응답 메시지 안에는 자원에 대한 정보와 자원을 가지고 있는 각각의 클라이언트에 대한 주소를 포함 한다 [1,2].

2.2. 기존의 수퍼 피어 기반 시스템

2.2.1 그누텔라 (Gnutella)

초기 그누텔라 0.4 버전에서는 순수 P2P 구조로써, 피어가 다른 피어와의 연결 정보를 얻기 위해서 Ping 메시지를 현재 연결된 모든 피어에게 전송한다. 수신한 피어들은 응답 값으로 자신의 연결 정보와 자원을 pong 메시지로 전송하며, 또한 수신한 ping 메시지를 자신과 연결된 다른 피어에게 브로드 캐스팅 한다. 따라서 각각의 노드들이 메시지를 핸들링하고, 라우팅 해야 하므로 노드의 수가 증가함에 따라 네트워크 내의 트래픽 또한 크게 증가된다. 그누텔라는 이러한 문제점을 해결하기 위해 버전 0.6에서는 수퍼 피어 개념인 ultrapeer를 사용한다. ultrapeer는 proxy로서 동작하는데, 노드들의 쿼리를 QRP(Query Routing Protocol)을 사용해서 대신 전달하는 역할을 한다. 그누텔라에서는 ultrapeer를 선출할 때 ultrapeer 자체의 cpu power와 network 성능에 몇 가지 제약사항을 둔다[3]. 그러나 그누텔라에서는 일반 노드와 ultrapeer 로 분류하여, 노드들은 power-law에 따라 ultrapeer를 랜덤하게 선택하게 된다. 따라서 성능이 우수한 ultrapeer의 경우, 많은 노드들의 쿼리를 대신 처리해 주어야 하기 때문에 병목현상이 발생할 수 있다. 또한 선출된 ultrapeer는 모두 동일한 성능을 가지고 있다고 가정하기 때문에 실제 일반 노드와 ultrapeer 사이의 성능에 대한 고려가 없다 [3,4].

2.2.2 FastTrack

FastTrack은 수퍼 피어 개념을 가장 먼저 사용한 시스템으로, FastTrack 에서는 성능이 좋은 노드들은 수퍼 노드로 자동적으로 선출되며, 일시적으로 더 낮은 성능의 클라이언트에 대해 index 서버의 역할을 제공한다. FastTrack은 초기 연결을 위해 서버에 프로그램 내에 수퍼 피어의 리스트를 저장한다. 따라서 일반 노드는 연결을 위해 활성화 된 수퍼 피어를 선택해서 수퍼 노드와 연결된다. 일반 노드는 자원을 검색하기 위해서 가장 가까운 위치에 있는 수퍼 노드에게 쿼리 메시지를 전송하며, 수퍼 노드는 대신 이를 처리한다.

수퍼 노드는 쿼리 메시지를 일반 노드에게 전송할 때 브로드캐스트(broadcast) 방식으로 전송하며, 자원을 다운로드 하는 방식은 그누텔라와 동일하다 [5].

3. 장 시스템 설계

3.1 개 요

제안된 시스템의 특징은 다음과 같다. 첫째, 일반 피어가 쿼리 메시지를 대신 처리해 줄 수 있는 최적의 수퍼 피어를 선택할 수 있도록 일반 피어 대 수퍼 피어의 성능을 분석한다. 이를 위해서 수퍼 피어와 일반 피어의 거리, 수퍼 피어의 응답 시간, CPU 이용률 등을 고려한다. 둘째, 수퍼 피어 자체의 쿼리 처리 능력을 분석한다. 이를 위해서 수퍼 피어가 일반 피어의 메시지 검색 요청에 얼마나 많은 결과 값을 전달했는지, 수퍼 피어가 가지고 있는 하드웨어의 성능, 일반 시간 동안 일반 피어의 메시지 검색 요청 쿼리 당 결과 값을 리턴해 주는 데 소용된 응답 시간 등을 고려한다. 제안된 시스템의 전체 구조는 그림 1과 같다.

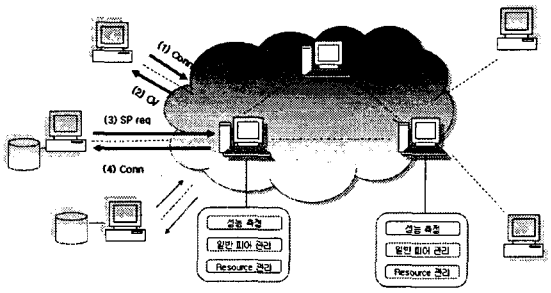

그림 1 ISP2P 시스템 설계
Fig. 1 ISP2P System Design

3.2 수퍼 피어의 성능 측정

먼저 일반 피어는 쿼리를 대신 전송해 줄 수퍼 피어를 선택한다. 수퍼 피어는 기본적으로 일반 피어 보다 우수한 성능을 가진 수퍼 피어이며, 시스템 사양의 최소 조건은 그누텔라에서 제시한 조건[3]을 만족해야 한다. 기존의 수퍼 피어 기반의 P2P 시스템에서는 수퍼 피어가 위와 비슷한 조건을 만족할 경우, 모두 동일한 수퍼 피어로 분류되었으며, 일반 피어는 쿼리 요청을 위해 랜덤하게 수퍼 피어를 선택하였다. 따라서 성능이 좋은 몇몇 수퍼 피어들은 일반 피어들에 의해 관계 요청을 집중적으로 받게 되며 과도한 쿼리 요청이 일어날 수 있다. 제안한 ISP2P 시스템에서는 이러한 문제점을 해결하기 위해 수퍼 피어들을 일반 피어와의 성능에 따라 분류하였다. 본 논문에서는 성능 측정의 기준으로 거리 비용, 응답 시간, CPU 처리율을 고려하였다. 거리 비용의 경우, 가장 최단 거리인 수퍼 피어를 선택하기 위한 기준이며, 응답 시간은 수퍼 피어와 일반 피어 사이의 네트워크 상태를 점검할 수 있는 기준이 된다. 수퍼 피어의 경우, 일반 피어를 대신하여 모든 쿼리를 처리해야 하므로 쓰레드 발생이 매우 높다. 따라서 수퍼 피어의 성능에 있어서 CPU 처리율은 중요한 요소가 된다.

- 거리 비용(Distance Cost): 링크 사이의 거리 비용은 전송 지연과 전파 지연을 값을 이용해 계산한다.

  - *전송 지연(Transmission delay) = Packet size / rate*

  - *전파 지연(Propagation delay) = Packet length / 2 * 10^8 to 3 *10^8 (meters/sec)*

  - *Total time = Transmission delay + Propagation delay*

- 응답 시간 (Response Time): 응답 시간은 수퍼 피어가 일반 피어의 연결 요청 메시지를 수신한 후 이에 대한 응답 메시지를 생성하여 전송하는 데 걸린 시간을 이용한다. 응답 시간을 통해 수퍼 피어와의 연결이 원활한지 살펴 볼 수 있다. 거리가 가까워도 네트워크 상태가 좋지 않을 경우 메시지의 전달과 처리에 문제가 발생될 수 있다. 제안한 시스템에서는 수퍼 피어가 네트워크상에서 얼

마나 안정적으로 일반 피어의 쿼리 메시지를 처리해주는지에 대한 정보를 제공하기 위해서 최근 과거의 평균 응답 시간을 이용하였다. 따라서 최근 과거의 평균 응답 시간과 현재 요청자 피어의 응답 시간에 관한 정보를 함께 이용하여 응답 시간을 계산하였다. 응답 시간은 현재 쿼리 응답 시간 대 과거 쿼리 응답시간으로 구한다. 먼저 과거 쿼리 응답시간은 지난 T 시간동안 일반 피어의 연결 요청 쿼리에 대한 응답시간을 구한다.

- *과거 쿼리 응답시간:*

$$\frac{1}{N}\sum_{i=0}^{N}((ctr(rmsg(s,q))) - (cts(smsg(q,s))))$$

- *응답 시간: 현재 응답 시간 / 과거 응답 시간*

- CPU 성능: CPU 성능은 현재 시스템에서 사용되고 있는 CPU의 이용량을 이용한다. 전체 CPU 사용량 μ (utilization)은 현재 사용되고 있는 프로세스의 이용률과 총 프로세스의 이용률을 사용해 계산한다.

- *Cμ (current utilization)= idle을 제외한 프로세스의 시간차의 합*
- *Tμ (Total utilization)= idle 프로세스의 시간차+ Cμ*
- *CPUμ (%) = Cμ/Tμ * 100*

3.3 성능을 고려한 수퍼 피어 선택

일반 피어가 수퍼 피어를 선택할 수 있도록 수퍼 피어의 전체 성능을 측정한다. 수퍼 피어의 성능은 수퍼 피어와 일반 피어 사이의 성능과 수퍼 피어 자체의 쿼리 처리 능력이다. 수퍼 피어 자체의 쿼리 처리 능력은 수퍼 피어가 피어의 쿼리에 대해 얼마나 짧은 시간에 많은 쿼리 결과 값을 피어에게 제공해 줄 수 있는지에 대한 척도이다. 수퍼 피어의 쿼리 처리 능력에 대해서 다음 3가지로 나누어 고려하였다.

- 최대 결과물 (Max Results): 지난 T 단위 시간 동안 수퍼 피어가 일반 피어의 메시지 검색 요청에 대해 얼마나 많은 결과 값
- 최대 성능 (Max Capacity): 수퍼 피어가 가지고 있는 자체 성능으로 수퍼 피어의 bandwidth, CPU, 하드 디스크의 크기, 메모리 사이즈에 대한 정보.
- 최대 속도 (Max Rate): 지난 T 시간 동안 일반 피어의 메시지 검색 요청 쿼리 당 결과 값을 리턴해 주는데 소요된 응답 시간

4. 장 시스템 구현

본 시스템은 일반 피어와 수퍼 피어로 나누어서 구현되었다. 구현 환경은 Window XP professional, Java 1.5 버전으로 Eclipse tool을 이용해 구현하였다. 현재 구현된 시스템은 일반 피어와 수퍼 피어의 성능 평가, 쿼리 전송이 원활한지 등을 살펴보기 위한 테스트 버전이다. 다음 그림 2는 일반 피어가 수퍼 피어를 검색한 결과로 최적의 수퍼 피어 순서로 정렬되어서 리스트에 보여 진다. 따라서 일반 피어는 가장 첫 번째 수퍼 피어에게 연결 요청 쿼리를 전송한다.

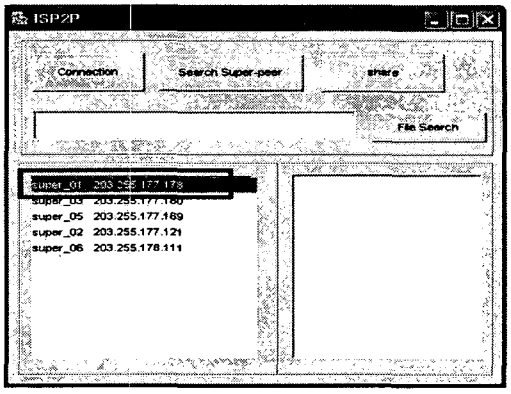

그림 2 수퍼 피어 검색
Fig. 2 Super-peer Retrieval

5. 장 결 론

본 연구에서는 ISP2P 시스템을 설계 및 구현을 통해 일반 피어의 성능 대 수퍼 피어의 성능을 분석해서 최적의 수퍼 피어를 선택할 수 있도록 제안하였다. 기존의 수퍼 피어 기반의 P2P 시스템에서는 모든 수퍼 피어를 동일하게 처리하였으며, 일반 피어는 단순히 랜덤하게 수퍼 피어를 선택하였다. 그러나 전체 P2P 시스템에서 일반 피어의 모든 쿼리를 대신 처리해야 할 수퍼 피어의 역할은 매우 중요하며, 이를 위해서 수퍼 피어를 성능에 따라 차별화 될 수 있도록 한다. 본 논문에서는 이를 위해 수퍼 피어의 성능 대 일반 피어의 성능을 분석해서 일반 피어가 최적의 수퍼 피어를 선택할 수 있도록 하였다. 수퍼 피어의 성능은 크게 2가지 부분으로 나누어 분석한다. 먼저, 일반 피어 대 수퍼 피어의의 성능을 측정하여, 수퍼 피어의 쿼리 처리능력에 대해 평가한다. 위의 두 내용을 기반으로 수퍼 피어의 성능을 차별화 한다. 따라서 일반 피어는 최적의 수퍼 피어를 통해 쿼리를 효율적으로 처리할 수 있도록 하며, 적절한 수퍼 피어의 선택으로 인해 전체 네트워크 성능- 대역폭, 응답 시간 등 또한 향상시킬 수 있다.

참 고 문 헌

[1] B. Yang, Garcia-Molina, "Designing a super-peer network", Tech., Stanford University. http://dbpubs.stanford.edu/pub/2002-13.

[2] Y. Chawathe , S. Ratnasamy , L. Breslau , N. Lanham and S. Shenker, "Making gnutella-like P2P systems scalable", *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, Karlsruhe, Germany, August 25-29, 2003.

[3] Gnutella protocol spec. v.0.6 http://rfc-gnutella.sourceforge.net /src/rfc-0_6-draft.html

[4] Gnutella website: http://www.gnutella.com

[5] KaZaA website: http://www.kazaa.com