

Peer-to-Peer 기반의 분산 파일 공유 기법

민수홍*, 조동섭
이화여자대학교 과학기술대학원 컴퓨터학과
e-mail: {shmin, dscho}@ewha.ac.kr

Distributed File Sharing based on P2P platform

Su-Hong Min*, Dong-Sub Cho
Dept. of Computer Science and Engineering, Ewha Womans University

요 약

최근 P2P 모델을 기반으로 한 애플리케이션의 등장으로 다양한 자원을 효율적으로 이용할 수 있게 되었다. P2P에서는 여러 대의 클라이언트를 상호 긴밀하게 연결함으로써 한 대의 서버에 다수의 클라이언트를 연결했을 때 보다 확실한 네트워크의 효과를 기대할 수 있다. 그러나, P2P 모델의 경우, 개인 PC가 서버로서의 역할을 동시에 수행하기 때문에 서버보다는 기능과 성능 면에서 떨어질 뿐 아니라 기존의 서버에 비해 상대적으로 낮은 처리 속도, 적은 메모리, 보안상의 문제점들이 있다. 본 논문에서는 P2P 기반의 문제점의 하나인 피어(Peer)간의 파일 공유로 인해 발생하는 문제들을 해결하고자 기존의 파일 공유 기법을 개선하였다. 제안된 파일 공유 기법은 데이터의 무결성을 보장하고, 파일의 상태 정보를 모니터링 함으로써 짧은 시간 안에 보다 효율적으로 파일을 교환할 수 있도록 하였다.

1. 서론

대부분의 인터넷 서비스는 전통적으로 분산된 클라이언트/서버 모델을 기반으로 하며, 현재 우리가 사용하는 대부분의 인터넷 애플리케이션(WWW, FTP, telnet 그리고 e-mail 등)들이 이 모델을 기반으로 하고 있다. 그러나 기하급수적으로 그 규모가 커져 가는 인터넷에 비해 기존의 클라이언트/서버 모델은 모든 서비스가 서버에 집중되어 있어, 인터넷의 자원인 정보, 대역폭, 컴퓨팅 자원을 활용하는데 있어 한계점이 있다. 이러한 문제를 해결하고자 P2P 모델이 등장하였다. P2P는 네트워크에 참여하는 모든 클라이언트가 서버의 역할을 동시에 수행한다. 따라서, 중앙 집중형 서버가 없는 환경에서 하나의 클라이언트에서 다른 클라이언트로 매우 다양한 경로를 통해 상호간 통신을 할 수 있다. 각각의 클라이언트들은 자신의 자원에 대한 접근을 다른 클라

라이언트에 허가함으로써 P2P 네트워크에 참여할 수 있도록 한다 [1,2]. 따라서, 인터넷을 통해 사용자들 간의 다양한 자원을 활용 할 수 있다. 그러나 P2P 모델의 경우, 불특정 다수에 의해 자원이 공유되기 때문에 공유되는 자원들에 대해 신뢰도가 낮은 문제점이 있다. 또한, 피어의 네트워크 상태 즉, 컴퓨터의 capacity, 네트워크 연결 상태 등에 대한 정보 부족으로 데이터 교환에 어려움이 따른다 [4]. 따라서, 본 논문에서는 이와 같은 문제점을 해결하기 위해 사용자에게 공유 자원의 무결성을 보장하며, 자원을 제공받고자 하는 피어들의 네트워크, 컴퓨터 capacity 정보를 서비스하고자 한다.

2. P2P 모델의 파일 공유 기법

P2P 모델은 서버가 존재하지 않는 순수한 P2P 모델인 분산형 모델과 디렉토리 서버를 둔 중앙집중형 모델로 나눌 수 있다. 이 장에서는 두 모델의 파일 공유 기법에 대해 알아보려고 한다.

이 논문은 2003년도 두뇌한국21사업에 의하여 지원되었음

2.1 분산형 모델 (Decentralized Model)

분산형 P2P 모델은 어떠한 중앙의 서버에도 의존하지 않고, 개개의 컴퓨터들에 의해서 동작한다. 일단 P2P 애플리케이션이 클라이언트의 메모리에 다운로드 되면, 피어들은 네트워크에 접속된 다른 피어들을 동적으로 찾으며, 연결된 피어들 사이의 모든 통신은 어떠한 서버의 도움도 받지 않는다. 피어가 콘텐츠에 대한 요청을 다른 피어에게 보낼 때, 원하는 콘텐츠가 이용 가능할 경우, 이 자원은 네트워크를 통해 다운로드 된다. 분산형 모델에서, 피어는 공유된 자원들을 보유하고 있는 서비스 제공자이다. 그러나 네트워크에 로그인한 피어들의 목록을 등록해 주는 중앙 관리자의 역할을 담당하는 것이 없기 때문에 사용자 스스로 다른 피어들의 위치를 찾아야 하는 문제점이 있다. 이와 같은 모델을 사용하는 애플리케이션으로는 그누텔라(Gnutella), 프리넷(Freenet) 등이 있다 [1,2].

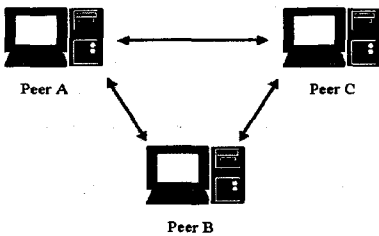


그림 1 분산형 모델

2.2 중앙집중형 모델(Centralized Model)

중앙집중형 모델은 클라이언트 상호간 효율적으로 통신하고 필요한 정보를 전달하기 위해 중간에 서버를 두는 방식이다. 클라이언트가 서로 통신을 하기 위해 최초로 IP 주소를 서버에 인덱스하며, 그 이후 서버의 도움 없이 클라이언트 상호간 직접 정보를 전달하게 된다. 파일을 공유하기 위해 사용자가 서버에게 질의를 전송하면, 서버는 이에 대한 응답으로 접속된 피어들의 목록과 파일의 위치 정보를 제공해 준다. 요청을 하는 피어는 요청한 파일의 목록에 근거해서 피어들에게 개별적으로 접근한다. 일단 요청을 하는 피어가 원하는 파일의 위치를 파악하면, 서버는 피어들의 연결을 개시하고, 피어는 네트워크를 통해서 원하는 파일을 다운로드 한다. 이 모델은 분산형 모델에 비해 피어들이 다른 피어들의 목록을 얻기가 쉬우며, 원하는 자원을 서버의 검색

기능을 통해 손쉽게 찾을 수 있어 편리하다. 이와 같은 방식을 사용하는 애플리케이션으로는 넷스터, 소리바다, MSN 메신저 등이 있다 [1,2].

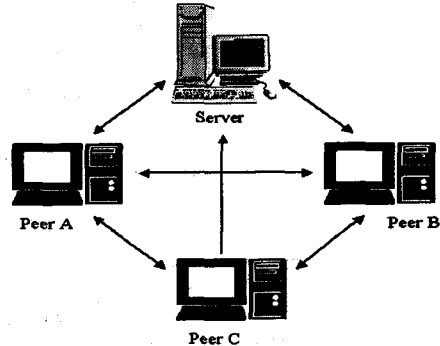


그림 2 중앙집중형 모델

3. 제안된 P2P 시스템 설계

이 장에서는 P2P 모델을 기반으로 한 전체 시스템을 설계하고, 본 시스템에서 사용되는 파일 공유 기법인 상태 정보 모니터링과 공유 파일의 무결성 보장에 대해 알아보려고 한다.

3.1 전체 시스템 구성도

본 논문에서 제시하는 P2P 모델은 중앙집중형 모델로써, 피어의 목록을 관리하는 서버를 가진 P2P 시스템이다. 서버가 없는 P2P 모델의 경우, 네트워크 트래픽 관리의 문제, 비인증된 접근이나 파일 다운로드와 같은 문제를 발생시킬 수 있다. 따라서 본 논문에서는 기존의 P2P 모델의 서버의 기능을 확장하여, 피어 간의 안전하게 파일을 공유할 수 있도록 하였다. 제시하는 P2P 모델은 서버, 리스너, 클라이언트로 구성된다 [2].

서버는 애플리케이션에서 필요로 하는 데이터베이스를 관리한다. 즉, 사용자의 IP 주소, ID, 패스워드와 관련된 로그인 정보와 사용자의 네트워크 상태 정보, 파일 정보 등을 관리한다. 리스너는 사용자의 요청을 처리하는데, 서버와 클라이언트 사이에서 요청을 질의하고, 수신하는 역할을 한다. 클라이언트는 사용자와 컴퓨터 사이에 인터페이스 역할을 한다. 사용자는 클라이언트를 통해 요청을 전송하고, 리스너의 응답을 수신한다. 또한, 서버에 접속해 있는 피어들의 목록을 보여주며, 피어가 요청한 결과물을

제시해준다.

다음 그림 3은 전체 시스템의 구조이다.

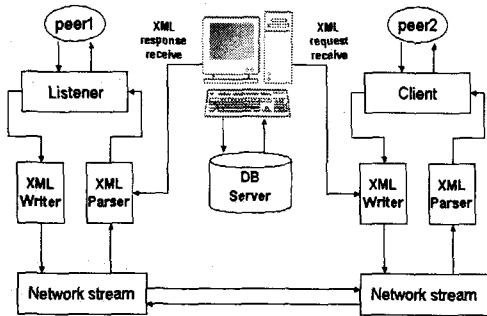


그림 3 전체 시스템 구조

3.2 파일 공유 기법

일반적으로 P2P 모델의 경우, 불특정 다수의 사람들이 네트워크를 통해 접근하고, 공유된 파일을 다운로드/업로드 하기 때문에 보안이나 서비스 질(QoS) 등에 관한 문제가 내재되어 있다 [4]. 예를 들어, 피어가 서버로 제공되는 원격 피어들의 파일에 대한 목록을 수신해서, 특정 피어에게 파일 서비스를 요청할 경우, 상대편 피어의 네트워크 상태 등에 따라 파일 전송 요청이 수락되거나 거절될 수 있다. 또한, 사용자가 요청하는 파일을 수신할 때 원본 데이터가 훼손된 파일이거나, 바이러스 등을 내포하는 파일 등을 다운로드받을 수 있다. 따라서, 본 논문에서는 이와 같은 문제점을 해결하기 위해 다음과 같은 두 가지 일을 수행하였다.

1) 피어 상태 모니터링 (Peer Status Monitoring)

상태 모니터링은 서버가 주기적으로 피어의 상태 정보를 수신하여, 이에 대한 정보를 요청하는 사용자에게 제공해준다. 사용자는 원하는 피어의 상태 정보를 모니터링 함으로써, 파일 전송에 대한 안전성을 보장받을 수 있다. 실제로 파일을 공유하는 피어들의 네트워크 상태는 가변적이기 때문에, 네트워크 상태에 따라 연결이 갑자기 끊어져 파일 전송이 중단될 수 있으며, 사용자가 이용하는 컴퓨터의 성능에 따라 파일 전송에 어려움을 겪을 수 있다. 따라서 본 논문에서는 피어들의 상태 정보를 서버가 주기적으로 수신하여, 이를 요청하는 피어에게 제공한다. 서버는 피어 간의 보다 안정된 파일 교환을 위해 피어의 네트워크 상태가 불안정하거나, 피어의 컴퓨터에서 바이러스가 발견될 경우, 피어의 연결을

강제적으로 해제하거나, 요청을 원하는 피어에게 이와 같은 정보를 제공한다 [3,5].

2) 공유 파일의 무결성 보장

현재 대부분의 P2P 애플리케이션의 경우, 파일에 대한 무결성을 보장할 수 없다. 실제로 서버를 포함한 P2P 애플리케이션들을 사용해 본 결과, 사용자가 원격지에 있는 피어들에게 다운로드받은 파일에 문제가 있는 경우를 종종 경험하게 된다. 예를 들어, 사용자는 다운로드받은 파일이 파일명과 불일치하거나, 훼손된 파일이거나, 바이러스 등에 감염된 것일 수 있다. 본 논문에서는 이 같은 문제를 해결하기 위해, 각각의 파일에 대한 메시지 다이제스트를 생성하여 파일의 무결성을 보장하고자 한다. 메시지 다이제스트를 생성하기 위해, 단방향 해쉬 함수인 MD5를 사용하였다. MD5는 임의의 길이의 메시지를 입력받아 128 비트 메시지 다이제스트를 생성하는 알고리즘으로, 주로 데이터의 무결성을 검증하는데 사용된다.

서버는 피어의 공유 파일의 정보를 보관할 때 파일명과 MD5 체크섬(checksum) 결과 값을 함께 보관하여, 피어가 클라이언트의 파일 검색을 통해 목록을 요청할 때 이를 토대로 파일명을 제공한다. 즉, 같은 체크섬 값을 가진 파일은 모두 동일한 파일이므로, 피어는 하나의 파일을 동시에 여러 피어들에게 요청할 수 있다. 따라서, 수신자는 보다 짧은 시간 안에 대용량의 파일을 전송 받을 수 있으며 [6], 송신자들은 전체 파일 중 일부분만 제공하게 되므로 피어들의 부하를 줄일 수 있다.

4. 시스템 구현

이 장에서는 제안한 파일 공유 기법을 이용해 P2P 시스템을 구현하였다.

4.1 시스템 환경

제안한 P2P 시스템은 다음과 같은 작업 환경에서 구현하였다.

표 1 시스템 환경

항목	시스템 환경
OS	MS Windows 2000 Adv.
Web Server	MS IIS(Internet Information Server)
DB Server	MS SQL 2000 server
Tool	JAVA SDK 1.4
Language	JAVA
Parser	Apache Xerces 1.4.4

4.2 서버 (Server)

서버는 웹서버와 데이터베이스 서버로 구성된다. 웹서버는 ASP(Active Server Page)를 사용해 리스너와 서버, 클라이언트와 서버 사이에서 통신한다. 데이터베이스 서버는 피어들의 로그인 정보와, 공유하는 파일의 정보를 관리하며, Peer 테이블과 Share 테이블로 구성된다.

Peer 테이블은 원격 IP 주소, 사용자명, 패스워드, 접속 상태 여부, 로그인 시간, 응답시간, 네트워크 트래픽을 저장한다. 다음 그림 4는 Peer 테이블의 구조이다.

이름	타입	길이	NULL	기본값
ip_address	char	20		
user_name	char	50		
passwd	char	20		
status	char	10		
connected_time	datetime	8		
response_time	datetime	8		
traffic	char	20		

그림 4 Peer 테이블

Share 테이블은 사용자의 원격 IP 주소, 공유 파일명, MD5 체크섬의 결과 값을 저장한다.

4.3 리스너 (Listener)

리스너는 서버에 로그인을 요청하고, 서버에 의해 인증이 허락되면, 리스너는 서버로부터 응답을 받는다. 리스너는 로그인 인증 여부를 확인하기 위해 이 응답을 XML 파서로 파싱하며, 리스너가 서버에 성공적으로 로그인되면, 현재 서버에 접속된 리스너의 목록에 해당 리스너의 항목이 추가된다. 서버에 로그인한 리스너는 자신에게 접속을 시도하고 요청을 전송하는 다른 피어들에게 응답을 처리한다 [2].

예를 들어, 서버가 소켓을 통해 피어의 상태 모니터링을 요청하면, 리스너는 이를 처리해서 XML 문서로 반환한다. 또한 리스너는 다수의 피어들의 요청을 처리하기 위해 쓰레드를 생성한다. 수신된 요청은 서버와 피어에 따라서 적절한 응답을 생성한다.

4.4 클라이언트 (Client)

사용자는 클라이언트 인터페이스를 통해 서버에 접근하며, 서버상에 존재하는 원격 피어들의 목록을

제공받는다. 또한 사용자는 클라이언트 인터페이스를 통해 자신의 공유하는 파일 정보를 제공하며, 원하는 파일을 검색기능을 통해 쉽게 찾을 수 있다. 또한, 상대 피어의 상태 정보를 요청할 수 있으며, 상대 피어의 상태 정보에 따라 원하는 파일을 여러 피어들로부터 동시에 요청하여 보다 빠른 시간 안에 제공받을 수 있다.

5. 결론

본 논문에서는 기존의 P2P 애플리케이션의 파일 공유 문제를 해결하기 위해, 피어의 상태 모니터링과 공유 파일의 무결성을 보장해주는 P2P 시스템을 설계 및 구현하였다. 우리가 제안한 P2P 시스템은 중앙집중형 피어 모델로서, 서버가 피어들의 네트워크 상태, 컴퓨터의 성능을 주기적으로 관찰하여 이를 피어들에게 제공해주며, MD5 알고리즘을 이용해 공유 파일의 무결성을 보장하였다. 따라서, 요청을 원하는 사용자는 원격 피어들의 네트워크 상태를 고려하여, 동일한 파일을 동시에 여러 피어들로부터 서비스 받을 수 있도록 하였다.

향후 연구로는 현재 연구한 것을 바탕으로 요청을 원하는 사용자에게 mining을 이용한 파일 검색 기법에 대해 연구하고자 한다.

참고문헌

- [1] Andy Oram, "Peer-to-Peer" O'Reilly, March 2001.
- [2] Dreamtech Software Team, "Peer-to-Peer Application Development", John Wiley & Sons, Nov. 2001.
- [3] Wilson, Brendon J, "JXTA", Macmillan Computer Pub, May 2002.
- [4] Manoj Parameswaran, Anjana Susarla, "P2P Networking: An Information-Sharing Alternative", IEEE Computer, July 2001.
- [5] Sang Li, "Early Adopter JXTA", Wrox Press Ltd. 2001
- [6] 김명선, 박인수, 박용진, "JXTA 프로토콜을 이용한 대용량 데이터 전송 서비스", 한국정보과학회 춘계 학술대회, 2002.