DHT 기반 P2P 시스템에서 키워드 검색 지원을 위한 시스템 디자인

진명희° 이승은 손영성† 김경석‡
부산대학교 컴퓨터공학과, ETRI†, 부산대학교 정보컴퓨터공학부‡
{mhjin°, selee, ysson, gimgs}@asadal.cs.pusan.ac.kr

System Design for Supporting Keyword Search in DHT-based P2P systems

Myounghee Jin° Seungeun Lee Youngsung Son Kyongsok Kim
Dept. of Computer Engineering, Pusan National University, Electronics and Telecommunications Research Institute, Division of Computer Science and Engineering, Pusan National University

요 약

분산 해시 테이블 (Distributed Hash Table) 을 사용한 P2P 시스템에서는 해시함수를 사용하여 파일과 노드의 ID를 정의하고 파일의 ID와 매핑 (mapping) 되는 ID를 가진 노드에 파일을 저장함으로써 시스템 전체에 파일을 완전히 분산시킨다. 이러한 시스템에서는 파일을 찾을 때 해시된 파일 ID로 찾기 때문에 정확한 매치 (exact match) 만 가능하다. 하지만 현재 P2P 파일 공유 시스템에서는 파일의 전체 이름을 정확히 알지 못하더라도 부분적인 키워드로 파일을 검색할 수 있도록 하는 키워드 검색 (keyword search) 이 요구된다. 본 논문에서는 분산 해시 테이블을 기반으로 하는 P2P 시스템에서 키워드 검색이 가능하도록 하는 방안을 제안한다.

1. 서 론

인터넷이 보급된 이후 지금까지 수많은 시스템이 클라이언트/서버 (Client/Server) 모델을 유지하고 있는 한편 P2P (Peer-to-Peer) 라는 새로운 패러다임이 출현하였다. 클라이언트/서버 환경에서는 서비스의 제공자와 사용자가 명확히 구분된다. 하지만 P2P 시스템에서는 모든 노드들이 서비스를 제공하기도 하고 제공된 서비스를 사용하기도 하면서 서버와 클라이언트의 역할을 동시에 담당한다. 이는 클라이언트의 컴퓨팅 능력이 향상되고 초고속 인터넷이 보급되면서 가능해질 수 있었다.

초기의 P2P 시스템에는 여전히 서버를 두고 서비스를 제공하는 Napster[1]와 서버를 두지 않고 방송 (broadcast) 을 통해 자원을 탐색하는 Gnutella[2]가 있다. Napster에서 서버는 직접적인 서비스를 제공하지 않지만 각 클라이언트의 파일 목록을 관리하기 때문에 단일 고장 지점의 원인이 될 수 있고 확장 가능성 (scalability) 에 제약을 가져올 수 있다. Gnutella는 자원 탐색을 위해 방송을 하기 때문에 아주 많은 네트워크 트래픽을 발생시킨다. 이 또한 시스템의 확장 가능성에 제약을 가져올 수 있다.

이후 P2P 시스템은 분산 해시 테이블을 기반으로 하여 체계적으로 구조화된 시스템으로 발전되어왔다. 이에 대한 대표적인 연구에는 Chord[3], Pastry[4], Tapestry[5] 그리고 CAN[6]이 있다. 이러한 시스템들은 분산 해시 테이블을 사용하여 파일을 완전히 분산시키고 효과적인 검색 서비스를 제공함으로써 초기의 P2P 시스템이 가진 확장 가능성의 문제점을 해결하고 있다. 또한 병목 현상과 단일 고장 지점의 원인이 되는 서버를 제거함으로써 시스템의 성능과 신뢰성을 향상시켰다. 특히 P2P 시스템의 특성인 노드의 잦은 접속과 단절에도 시스템이 잘 견딘다는 점은 이 시스템들이 제공하는 큰 장점이다. 하지만 이러한 장점들에도 불구하고 현재 분산 해시 테이블을 기반으로 하는 시스템들은 Napster나 Gnutella가 제공하는 키워드 검색 기능을 제공하지 못하고 있다. 본 논문에서는 분산 해시 테이블을 기반으로 하는 P2P 시스템에서 각 노드들이 유지하는 파일에 대한 키워드의 인덱스를 유지함으로써 키워드 검색이 가능하도록 하는 방안을 제안한다.

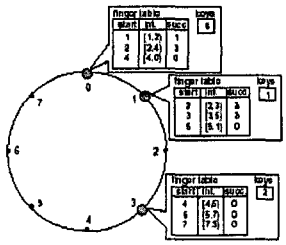
본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 살펴보고 3장에서는 본 논문의 핵심인 키워드 검색을 위한 시스템 디자인에 대해 살펴본다. 마지막으로 4장에서는 결론과 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 Chord

Chord는 분산 해시 테이블을 기반으로 하는 확장 가능한 P2P 탐색 프로토콜이다. Chord는 successor(key) 라는 주어진 key에 대해 key의 successor 노드를 리턴하는 하나의 연산만 제공한다. successor는 ChordID 공간에서 key와 같거나 key의 위치에서 시계방향으로 가장 인접해 있는 노드를 가리킨다. 이 연산을 반복함으로써 원하는 파일을 검색할 수 있다. key와 노드의 ID는 160 비트 SHA-1 해시 함수를 통해서 할당된다. 시스템에 참여한 각 노드의 라우팅 테이블은 [그림 1]과 같다. 이 라우팅 테이블은 다른 노드들에 대한 약간의

정보만으로 구성되며 테이블의 크기는 O(logN)이다. 또한 Chord는 O(logN)의 자원 탐색 비용을 보장함으로써 확장 가능성을 제공한다[3]. 본 논문은 Chord를 기반으로 하고 있다. 따라서 파일을 저장하고 검색하는 부분에서 Chord의 라우팅 기법을 그대로 사용한다.



[그림 1] 노드 0, 1 그리고 3에 대한 라우팅 테이블

2.2 키워드 검색

키워드를 n-gram[7]으로 나누면 검색에 별로 도움이 되지 않는 키워드 까지도 저장이 될 수 있고 저장 공간도 많이 요구된다. 따라서 본 논문에서는 키워드를 공백으로 구분해서 저장함으로써 저장 공간을 줄인다. 보통 P2P 파일 공유 시스템의 키워드 검색은 완벽하게 정확한 결과만을 요구하지 않는다. 따라서 본 논문에서는 키워드가 2개 이상일 경우 [8]과 같이 공통부분(intersection)을 구하기 위한 연산을 하지 않고 처음 검색을 요청한 노드에게 각 키워드에 대한 파일 ID를 모두 보낸다. 그리고 결과를 받은 노드는 사용자에게 결과를 정렬해서 리턴 한다. 이때 공통으로 키워드가 존재하는 파일에 우선순위를 둔다.

3. 시스템 디자인

이 장에서는 키워드 검색이 가능하도록 하기 위해 파일이 시스템에 저장될 때 파일에 대한 키워드의 인덱스가 어떻게 구성되는지와 그 인덱스를 사용하여 키워드로 파일을 검색하는 기법에 대해 살펴본다.

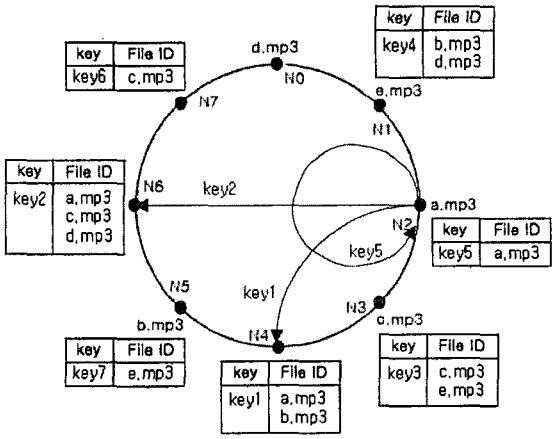
3.1 파일 저장

파일이 시스템에 저장되면서 각 파일에 대한 키워드의 인덱스는 [그림 2]와 같이 분산 해시 테이블을 통해 각 노드로 완전히 분산된다. [그림 2]의 시스템에는 N0~N7까지 8개의 노드가 존재하고 [표 1]에 있는 5개의 파일이 시스템에 저장되어있다. 파일이 저장되는 방법은 기존 Chord와 마찬가지로 해시된 파일 ID에 매핑되는 ID를 가진 노드에 저장된다[3]. 키워드에 대한 인덱스는 파일이 저장될 때 분산 해시 테이블을 통해 각 노드로 분산되는데 [표 1]에 있는 각 파일에 대한 키워드의 인덱스가 [그림 2]와 같이 시스템에 존재하는 노드들에 분산되어 있다.[1)]

파일에 대한 키워드는 160 비트 SHA-1 해시 함수를 사용하여 ChordID 공간에 매핑 된다. (키워드는 파일 이름이 될 수도 있고 사용자가 정의한 파일에 대한 정보가 될 수도 있다.) 이 때 매핑된 노드에는 해시된 key와 파일 ID를 저장함으로써 인덱스를 구성한다. 예를 들면 [표 1]에서 파일 ID가 a.mp3인 파일의 키워드는 key1, key2 그리고 key5 이므로 세 개의 키워드를 각각 해시한 값과 매핑되는 노드가 N4, N6 그리고 N2라고 했을 때 [그림 2]와 같이 해시된 key와 파일 ID가 해당되는 노드의 인덱스에 저장된다.[2)]

[표 1] 시스템에 저장된 파일에 대한 키워드

File ID	Keywords
a.mp3	key1, key2, key5
b.mp3	key1, key4
c.mp3	key2, key3, key6
d.mp3	key2, key4
e.mp3	key3, key7



[그림 2] 시스템에 저장된 파일에 대한 키워드의 인덱스

3.2 파일 검색

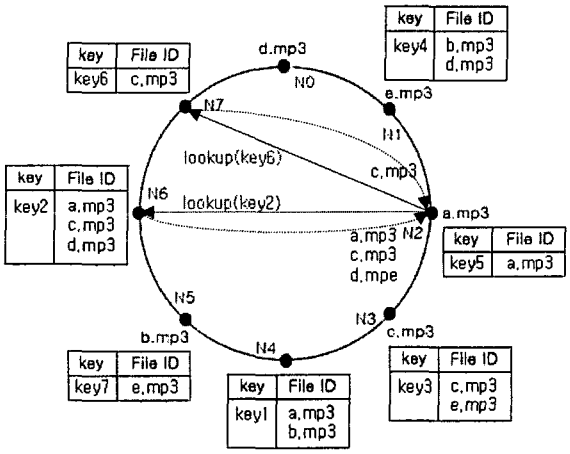
원하는 파일을 검색하기 위해서는 파일의 전체 이름을 알지 못하더라도 파일에 대한 부분적인 키워드만 가지고 파일을 찾을 수가 있다. 키워드가 하나일 경우에는 Chord에서 제공하는 정확한 매치 (exact match) 로 파일을 검색할 수 있다. 키워드가 두 개 이상일 경우에는 각 키워드로 탐색을 하여 해당되는 모든 파일 ID를 구한 다음 파일 ID 리스트를 정렬해서 사용자에게 리턴 한다. 그러면 사용자는 리스트에서 원하는 파일을 선택하여 다운로드 할 수 있다. 아래에 간단한 예제를 살펴보자.

노드 N2가 원하는 파일이 [표 1]에 있는 c.mp3이고

1) 물론 모든 노드는 하나 이상의 key에 대한 인덱스를 유지할 수 있지만 [그림 2]에서는 하나의 key에 대한 인덱스만 나타냈다.

2) [그림 2]에서 Chord 시스템의 라우팅 테이블은 생략했다.

이 파일에 대해 알고 있는 키워드가 key2와 key6 뿐이라고 했을 때 파일 검색 방법은 [그림 3]과 같다.



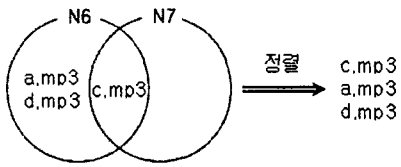
[그림 3] c.mp3 파일 검색 과정, 알고 있는 키워드는 key2와 key6

1) 노드 N2는 알고 있는 두 개의 키워드 key2와 key6을 각각 해시한다.

2) key2와 key6을 해시한 값에 해당되는 노드가 각각 N6과 N7이라면 [그림 3]과 같이 노드 N2는 두 개의 노드에 메시지를 보내서 각 key에 대한 파일 ID를 요청한다.

3) 메시지를 받은 N6과 N7은 각각 key에 대한 파일 ID를 노드 N2에게 응답한다. key2에 대한 파일 ID는 a.mp3, c.mp3 그리고 d.mp3이고 key6에 대한 파일 ID는 c.mp3이다.

4) 응답 메시지를 받은 노드 N2는 [그림 4]와 같이 각 key에 대한 파일 ID의 공통부분을 추출하여 우선순위를 부여한 다음 정렬해서 사용자에게 결과를 리턴 한다.



[그림 4] 리턴 결과 리스트 정렬

5) 사용자가 리턴 된 파일 리스트에서 파일을 선택하면 파일 ID를 해시하여 실제 파일이 위치한 노드와 P2P로 연결되면서 파일을 다운로드 받을 수 있게 된다. 만일 c.mp3을 선택한다면 노드 N3과 연결될 것이다.

4. 결론

분산 해시 테이블을 이용한 P2P 시스템이 각광받는 여러 가지 이유 중 하나로 시스템의 확장 가능성을 들 수 있다. 이에 따라 Gnutella에서 시스템의 확장 가능성을 제공하도록 제안한 논문[9]이 있는데 확장 가능성이 제공되는 분산 해시 테이블 보다는 Gnutella를 확장 가능한 시스템으로 발전시키는 것이 낫다는 주장을 펼치고 있다. 그 이유로 몇 가지를 들고 있는데 그 중 하나가 분산 해시 테이블이 정확한 매치만 가능하고 키워드 검색이 되지 않는다는 것이다. 따라서 본 논문에서는 분산 해시 테이블을 이용한 P2P 파일 공유 시스템에서 키워드 검색이 가능하도록 하는 기법을 제안하였다. 시스템에 파일을 저장할 때 파일에 대한 키워드의 인덱스를 전체 시스템에 완전히 분산시키는 방법으로 저장 공간의 낭비를 줄일 수 있다. 또한 인덱스가 분산됨에 따라 네트워크 부하를 분산시키는 효과를 얻을 수 있다.

향후 노드나 파일이 시스템에 더 이상 존재하지 않게 될 경우 인덱스를 수정하고 그 일관성을 유지하는 방안에 대한 연구가 필요하다. 그리고 인기가 높은 키워드로 집중되는 네트워크 트래픽을 분산 시키는 방안에 대한 연구도 요구된다.

5. 참고문헌

[1] Napster, http://www.napster.com
[2] Gnutella, http://www.gnutella.com
[3] I. Stoica, R. Morris, D. Karger, F. Kaashoek and H. Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, ACM SIGCOM'01, Aug. 2001
[4] A. Rowstron and P. Druschel, Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, IFIP/ACM International Conference on Distributed Systems Platforms, Nov. 2001
[5] B. Zhao, J. Kubiatowicz and A. Joseph, Tapestry: An infrastructure for fault-tolerant wide-area location and routing, Tech. Rep. UCB/CSD-01-1141, Computer Science Division, Univ. of California, Berkeley, Apr. 2001
[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, A Scalable Content-Addressable Network, ACM SIGCOMM'01, Aug. 2001
[7] M. Harren, J. M. Hellerstein, R. Huebsch, B. T. Loo, S. Shenker and I. Stoica, Complex Queries in DHT-based Peer-to-Peer Networks, In Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems(IPTPS'02), Mar. 2002
[8] P. Reynolds and A. Vahdat, Efficient Peer-to-Peer Keyword Searching, Tech. Rep. Duke University, CS Department, Feb. 2002
[9] Y. Chawathe, S. Ratnasamy, L. Breslau, Making Gnutella-like P2P Systems Scalable, ACM SIGCOMM'03, Aug. 2003