

# DHT 기반 P2P 시스템에서 자율적인 인기도 측정을 이용한 복제 전략

한동윤<sup>o</sup> 차봉관 손영성 김경석

부산대학교 컴퓨터공학과 데이터베이스 및 인터넷컴퓨팅연구소  
{dyhan<sup>o</sup>, bgcha, ysson, gim2005}@asadal.cs.pusan.ac.kr

## Replication Strategy using autonomous measurement of popularity on DHT based P2P System

Dongyun Han<sup>o</sup> Bonggwon Cha Youngsong Son Kyongsok Kim  
Pusan National University Database & Internet Computing Laboratory

### 요 약

P2P (Peer-to-peer) 파일 공유 시스템에서 콘텐츠의 복제는 Hot Spot의 방지와 가용성의 보장, 검색 속도의 향상을 위해 사용되고 있다. 콘텐츠의 복제본의 수(공간 효율성)와 검색 성능은 일종의 trade-off로써 이 논문에서는 가능한 검색 성능을 향상시키면서 적절한 양의 복제본의 수를 유지하는 방법인 SRoD 알고리즘을 제안한다. SRoD는 DHT 기반 파일공유시스템에서 전반적으로 간단하게 적용하여 사용할 수 있다는 장점을 가지고 있다. 이러한 특징은 P2P가 추구하는 자율성과 평등성을 보장하기 위한 것이다. 본 논문에서는 SRoD 알고리즘을 설명하고 이것의 장점과 보완해야 할 점에 대해 논하게 될 것이다.

### 1. 서 론

인터넷의 발달과 더불어 네트워크 기반 장치의 발전으로 이것을 이용한 여러 가지 시스템들이 개발되고 있다. 그 중에서도 최근에 가장 널리 사용되고 있는 시스템 중의 하나가 바로 P2P 파일공유시스템이다. 많은 형태의 P2P 파일공유시스템이 이용되고 있고 연구되고 있으며 그 중에서도 애플리케이션 레벨에서 간단한 구조를 가지면서도 좋은 검색 성능과 함께 자율성, 평등성, 확장성을 보장하는 DHT (Distributed Hash Table) 기반의 P2P 파일공유시스템에 대한 연구가 활발하게 이루어지고 있다.

가장 대표적인 DHT 기반 P2P 파일공유시스템으로는 Chord, CAN, Pastry, Tapestry 등이 있다 [1, 2, 3, 4]. 이 시스템들의 공통적인 특징은 시스템에 참여하는 모든 피어들이 자율적으로 일정 수의 피어들에 대한 라우팅 정보만을 유지하고 있으면 간단한 라우팅 알고리즘에 의해 일정 수 이하의 피어만을 거쳐서 모든 파일에 대한 검색이 가능하다는 것이다.

P2P 파일공유시스템에서 콘텐츠의 복제는 Hot Spot의 방지와 가용성의 보장, 검색 속도의 향상을 위해 사용되고 있다. Hot Spot이란 특정 피어에 지나치게 많은 요청이 들어와서 그 주변의 네트워크에서 정체 현상이 발생하는 것을 의미한다. 물론 가능한 한 많이 콘텐츠를 복제하게 되면, 예를 들어 모든 피어가 시스템에 삽입되는 모든 콘텐츠를 복제해서 가지고 있다고 한다면, 전체 시스템에서 검색 성능은 최대로 향상될 수 있을 것이다. 그러나 시스템 전체의 저장 공간은 한정되어 있으므로 불필요한 콘텐츠의 복사로 인한 저장 공간의 낭비는 오히려 P2P 파일공유시스템의 가장 기본적인 목적인 네트워크를 이용한 효율적인 대용량 저장 공간의 구축을 방

해할 수 있다. 이를 위해 콘텐츠의 인기도에 따라 전체 시스템에 들어갈 효율적인 복제본의 개수를 결정해 주는 여러 가지 수학적 방법론들이 제시되고 있다 [5, 6]. 이 연구들은 적절한 복제본의 개수가 콘텐츠 사이의 인기도 비율에 의해 좌우되지만 저장 공간의 효율을 위해서는 콘텐츠들의 복제 비율은 인기도 비율의 제곱근에 비례하는 것이 좋다는 것을 수학적으로 증명해 놓았다. 그러나 이 방법론들을 적용하려면 주기적으로 전체 아이템의 인기도를 계산하고 복제 명령을 내리기 위한 추가적인 제어장치가 필요하게 되고 이것은 P2P의 본질적인 요소인 평등성과 자율성을 침해하게 된다. 그리고 이 과정에서 네트워크에 불필요한 부하를 증가시킬 수 있다. 그래서 이 논문에서는 이러한 평등성과 자율성을 침해하지 않으면서 효과적으로 복제할 수 있는 방법인 SRoD (Self-controlled Replication strategy on DHT based peer-to-peer system) 을 제시하고 있다.

이 논문은 다음과 같이 구성된다. 먼저 2장에서는 본 논문에서 제시하는 복제 전략을 위한 자료 구조에 대해 설명할 것이다. 그리고 3장에서는 그 구조를 이용한 복제 전략에 대해 설명하고 그 효율성을 대략적으로 증명할 것이다. 4장에서는 결론과 함께 보완해야 할 점에 대해 이야기하게 될 것이다.

### 2. SRoD를 위한 자료 구조

SRoD를 위한 자료 구조는 간단하다. 본 논문에서 제시하고자 하는 복제 전략의 핵심은 자율적이고 단순한 구조를 만들고자 하는 데 있으므로 그것을 위한 자료 구조 역시 단순해야 한다.

그림 1에서 보듯이 전체 구조는 Replicated Space와

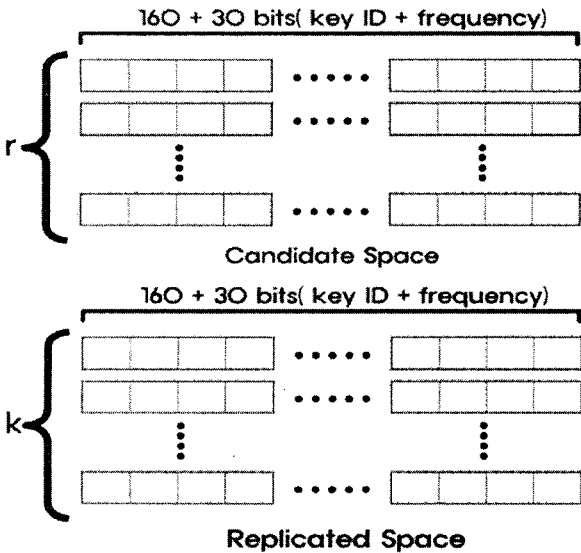


그림 1. SRoD를 위한 자료 구조

Candidate Space로 구성되며 각각 시스템에서 정의된 개수만큼의 key ID와 빈도를 저장할 공간을 가지게 된다. 일반적으로 DHT기반의 P2P 시스템에서 key ID의 크기는 160bits이고 [1, 2, 3, 4] 빈도를 저장할 공간을 30bits로 두면 Candidate Space의 크기는  $190 * k$  bits가 되고 ( $k$  : candidate space에 저장할 콘텐츠의 개수) Replicated Space의 크기는  $190 * r$  bits ( $r$  : 각 노드에 저장할 수 있는 콘텐츠의 개수)가 된다. 그리고 threshold 값의 저장을 위한 30bits의 저장 공간이 필요하다.

두 Space는 새로운 lookup 메시지를 받을 때마다 갱신되며 각 Key ID의 빈도에 따라 내림차순으로 정렬한다. 자세한 갱신 과정은 다음 장에서 설명할 것이다. 그리고 주기 T마다 두 space에 저장되어 있는 모든 key에 대한 frequency는 1씩 감소시킨다. 이렇게 하는 이유는 일반적으로 콘텐츠의 인기는 시간이 지나면 감소하게 마련인데 이러한 경향을 적용함으로써 불필요한 공간의 낭비를 막기 위함이다. 그리고 이 두 space에 피어의 Node ID와 동일한 Key ID는 저장될 수 없다. 그 이유는 자신의 Node ID와 같은 Key ID를 가진 콘텐츠는 당연히 자신이 관리해야 하기 때문이다.

### 3. SRoD 알고리즘

#### 3.1 Lookup 메시지를 받았을 때

어떤 피어가 특정 Key ID에 대한 Look 메시지를 수신하면 먼저 그 콘텐츠가 자신의 피어에 복제되어 있는지 알아봐야 한다. 그러기 위해 먼저 Replicated Space에 그 Key ID가 있는지 살펴본다. 만약 존재한다면 Lookup 메시지를 보낸 피어에게 그 Key ID에 해당하는 콘텐츠를 가지고 있다는 응답 메시지를 전송하게 된다. 그리고 Replicated Space에 있는 해당 Key ID의 빈도를 1만큼

증가시키고 윗 행에 있는 Key ID의 빈도와 비교하여 크다면 자리를 교체하게 된다. 만약 Replicated Space에 그 Key ID가 존재하지 않는다면 routing table을 이용한 정상적인 DHT 기반 P2P 시스템의 Lookup 과정이 계속 되게 될 것이다. 그리고 Key ID는 Candidate Space에 저장되며 그것이 이미 Candidate Space에 저장되어 있는 Key ID와 같다면 그것의 빈도만 1만큼 증가시키게 되고 바로 위에 있는 Key ID의 빈도와 비교하여 새로 증가된 빈도가 더 크다면 자리를 맞추게 된다. 이것은 빈도가 같거나 더 큰 Key ID를 만날 때까지 반복된다. 이렇게 함으로써 자연스럽게 Candidate Space는 정렬된 상태를 유지하게 된다. 만약 같은 Key ID가 존재하지 않는다면 그 Key ID는 Candidate Space에 저장된다. 그리고 이 경우 Candidate Space에 더 이상의 저장 공간이 없다면 가장 빈도가 적은 Key ID와 교체하게 된다.

#### 3.2 복제 여부의 결정과 삭제

복제 여부를 결정하기 위한 threshold는 시스템 관리자가 피어의 네트워크 전송 능력과 콘텐츠의 용량 등을 고려하여 결정하게 된다. 이것을 결정하는 방법은 상황에 따라 달라지며 각 피어마다 다르게 할 수도 있고 동적으로 변동이 가능하게 할 수도 있다. 그러나 이 논문은 모든 피어가 동일한 능력을 가지고 있고 모든 콘텐츠의 크기를 동일하다고 가정하는 일반적인 DHT 기반의 P2P 파일공유시스템을 고려하고 있으므로 threshold 역시 일정하다고 가정한다.

Candidate Space의 각 Key ID의 빈도는 이전에 설명했던 것처럼 그 Key ID에 대한 메시지를 받으면 증가하고 주기적으로 1씩 감소하게 된다. 이런 과정을 거치다가 어떤 Key ID의 빈도가 threshold와 같아지면 이 Key ID와 빈도 정보는 Candidate Space에서 삭제되어 Replicated Space로 이동하며 그 Key ID에 대한 전송 메시지를 보내어 해당하는 콘텐츠를 다운받아서 저장하게 된다. 이 때 Candidate Space의 두 번째 행과 그 아래에 있던 Key ID와 빈도 정보들은 위로 한 칸씩 이동하게 된다.

하지만 이 경우 Replicated Space가 가득 차 있다면 즉 더 이상 콘텐츠를 복제할 수 없는 상태라면 일단 Candidate Space에서 삭제하지 않고 빈도를 Replicated Space의 가장 아래에 있는 Key ID의 빈도와 비교하게 된다. 이 때 복제하고자 하는 Contents의 Key ID, 즉 Candidate Space에 있는 Key ID의 빈도가 크다면 서로 자리를 바꾸게 되고 Replicated Space에 있던 Key ID에 해당하는 콘텐츠는 삭제하고 새로운 Key ID에 해당하는 콘텐츠를 다운받게 된다. 그리고 새로 Replicated Space로 이동한 Key ID와 빈도 정보는 위에 있는 행들과 빈도를 비교하여 자신보다 큰 빈도를 가진 Key ID를 만날 때까지 계속해서 자리를 이동하게 된다. 그러나 만약 Replicated Space에 있는 Key ID의 빈도가 큰 경우에는 Candidate Space에 있던 Key ID는 대기하고 있다가 계속해서 메시지가 들어와서 빈도가 Replicated Space의 최하위 행에 있는 빈도보다 커지게 되면 비로소 Replicated Space에 들어갈 수 있다.

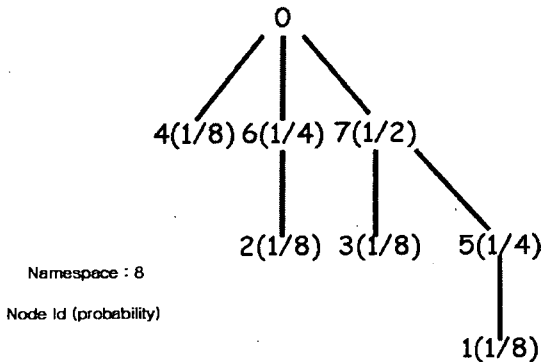


그림 2. namespace가 8인 Chord 시스템에서 모든 노드가 참여하고 있을 때, 노드 ID가 0인 피어를 향해 routing 할 때의 경로.

그리고 주기적으로 두 space에 있는 모든 빈도를 1씩 감소시켜서 Replicated Space에 있는 Key ID의 빈도가 threshold의 50%가 되면 Candidate Space로 이동하고 해당하는 컨텐츠는 삭제하며 Candidate Space를 새로 정렬시킨다. 그리고 Candidate Space에 있는 Key ID의 빈도가 0이 되면 그 정보는 Candidate Space에서 삭제하게 된다. 이렇게 하는 것은 저장 공간의 효율을 높이기 위한 방법이기도 하지만 너무 자주 복제된 컨텐츠를 삭제하거나 다운받지 않음으로써 불필요한 네트워크 부하를 줄이기 위한 것이다.

### 3.3 DHT 기반 P2P 시스템의 Routing 경로 분석

DHT 기반 P2P시스템에서 하나의 특정 피어에 대한 Routing 경로를 분석해 보면 그것이 트리 구조와 같다는 것을 알 수 있다. 그림 2는 Chord에서 노드 0을 찾는 메시지의 라우팅 경로를 나타낸 그림이다. 그림에서 각 노드의 앞의 숫자는 피어의 노드 ID를 나타내며 괄호 안의 분수는 그 피어를 지나갈 확률을 의미한다. Namespace가 커지게 되면 이 트리도 똑같이 확장된다고 보면 된다. 그러나 모든 노드가 시스템에 참여하지 않는 경우 참여하지 않은 노드의 정보를 관리하는 노드는 트리의 부모 노드가 아닌 경우가 대부분이므로 어떤 노드가 시스템이 있느냐에 따라 그 확률분포는 다양한 형태를 띠게 된다. 이전의 연구에서는 위 그림과 같은 특성에 착안하여 각 피어의 일정 수의 predecessors (ID 상으로 바로 앞이 있는 노드들, 이 예에서 0의 3개의 predecessors는 7, 6, 5) 에 복제하는 방안을 제안하기도 하였다. [7] 하지만 이 때 복제 개수를 3이라 가정하고 5번 노드가 없다면 그 부하는 6번 노드에 적용이 되는데 별 의미가 없는 4번 노드에도 복제를 하여야 한다. 이런 경우는 빈번하게 발생할 수 있다. SRoD는 threshold값을 조정하는 것만으로 그림 2의 분석에서 나온 결과를 잘 반영하는 것은 물론 어떤 상황에서도 의미 없는 복제를 하지 않는다.

### 4. 결론 및 향후 연구 방향

본 논문에서 제시한 SRoD 알고리즘은 Hot Spot을 예방하기 위한 복제 전략으로 DHT 기반 P2P 파일공유시스템에 전체적으로 적용하여 사용할 수 있다. 또한 P2P 고유의 자율성과 평등성을 보장하며 많은 저장 공간을 사용하지 않고 불필요한 메시지를 전송하지 않는 단순하고 부하가 적은 알고리즘이다.

그러나 복제 전략을 사용하는 가장 중요한 이유 중의 하나인 가용성 문제를 고려할 때 SRoD는 명확한 약점을 가지고 있다. 가령 어떤 피어가 갑자기 다운되었을 때 그 피어에서 관리하고 있던 컨텐츠를 원하는 사용자는 복제된 컨텐츠를 사용할 수 있어야 하지만 SRoD는 복제된 컨텐츠가 그 사용자로부터 다운된 피어를 향하는 경로 위에 놓여 있지 않은 이상은 사용할 수 없다. 예를 들어 그림 2와 같은 구조에서 노드 ID가 0인 피어에 있는 컨텐츠가 노드 ID가 7인 피어에 복제되어 있을 때 1, 3, 5번 피어의 사용자들은 이 컨텐츠를 사용할 수 있지만 2, 4, 6번 피어의 사용자들은 이 컨텐츠를 사용할 수 없게 된다.

앞으로는 이러한 가용성을 높일 수 있는 방안에 대한 연구와 더불어 실질적인 구현을 통해 SRoD의 효율성을 더욱 재고할 수 있는 방안을 연구해 보고자 한다.

### 관련 논문

[1] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Krishnan.  
"Chord: A Karger, M. Frans Kaashoek, Frank Dabek, and Hari Bala Scalable Peer-to-Peer Lookup Protocol for Internet Applications"  
IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 11, NO. 1, FEBRUARY 2003

[2] Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker.  
"A Scalable Content-Addressable Network,"  
SIGCOMM, 2001.

[3] ROWSTRON, DRUSCHEL.  
"Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems."  
In Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001) (Nov. 2001)

[4] Y. Zhao, J. D. Kubiawicz, and A. D. Joseph.  
"Tapestry: An Infrastructure for Fault-Resilient Wide-area Location and Routing."  
Technical Report UCB//CSD-01-1141, U. C. Berkeley, April 2001.

[5] Edith Cohen, Scott Shenker.  
"Replication Strategies in Unstructured Peer-to-Peer Networks"

[6] Jussi Kangasharju, Keith W. Ross, David A. Turner.  
"Optimal Content Replication in P2P Communities"

[7] Min Cai, Ann Chervenak, Martin Frank  
"A Peer-to-Peer Replica Location Service Based on a Distributed Hash Table"