

# DHT 기반 P2P 시스템에서의 범위 검색을 위한 시스템 설계

한중욱\*, 염대영\*, 한동윤\*, 김정석\*\*

\*부산대학교 컴퓨터공학과

\*\*부산대학교 정보컴퓨터공학부

e-mail:{jwhan,dyyoum,dyhan,gimng}@asadal.cs.pusan.ac.kr

## System Design for Range Search on P2P System based DHT

Jong-Wook Han\*, Dea-Young Youm\*, Dong-Yun Han\*, Kyung-Suk Kim\*\*

\*Dept of Computer Engineering, Pusan University

\*\*Dept of Computer Science and Engineering, Pusan University

### 요 약

인터넷의 보급과 PC의 발달로 정보의 분산과 공유가 가속화를 이루었고, 컴퓨팅 구조가 인터넷 기반  
으로 변화하였다. 이후 지금까지 수많은 시스템들이 클라이언트/서버 모델을 유지하고 있는 한편 대단  
위 분산 다중서버로 구성되는 환경이 출현하였다. 클라이언트/서버 모델에서 사용자들의 정보 공유의  
제약을 해소하기 위해 개인의 피어 사이에 서로 공유하는 P2P(Peer-to-Peer) 패러다임이 출현 되었다.  
하지만 기존의 DHT(Distributed Hash Table)를 기반으로 하는 P2P 시스템에선 정보를 검색할 때, 검색  
어와 검색결과와의 모호성이 존재하고, 범위 검색을 지원하지 못하는 문제가 있었다. 우리는 이 문제를  
해결할 한 가지 방안을 본 논문에서 제안한다.

### 1. 서론

인터넷의 보급과 함께 정보는 서버에 축적되어 클라이  
언트가 이용하는 방식인 클라이언트/서버 모델의 시스템  
이 주를 이루었다. 하지만 전세계 사용자들이 만들어 내는  
엄청난 양의 정보가 다양한 미디어에 저장되고, 이를 공유  
하기 위한 새로운 패러다임이 필요하게 되었다. 이로 인해  
각 피어들이 클라이언트이자 서버의 역할을 하는 P2P 시  
스템이 출현하였다.

초기의 P2P 시스템은 중앙에 서버를 두고 서비스를 제  
공하는 Napster[1]와 같은 방식이었다. 하지만 이는 중앙  
서버의 성능 저하가 전체 시스템의 성능 저하로 이어지는  
문제가 있다. 이후 플러딩(flooding) 방식을 이용하였으나,  
이는 질의할 때 시스템 전체에 많은 부하를 초래하는 문  
제가 있었다. 현재 이런 문제점을 해결하고 평균  $O(\log N)$   
의 성능을 보장하는 DHT 기반의 P2P시스템들을 많이 등  
장하였다. 대표적으로 CAN[2], Chord[3], Pastry[4],  
Tapestry[5]가 있다. 이 시스템들은 물리적인 위치를 고려  
하지 않았거나, 고려했다라도 피어의 성능을 고려하지 않  
았다. 이런 문제점을 해결하기 위해 MasicSquare[6]가 고  
안 되었고, 이를 본 논문에선 기반 시스템으로 사용한다.

DHT 기반의 P2P 시스템들은 검색의 모호성(Semantic  
Gap)으로 인해 사용자가 원하는 정보 이외에 필요 없는  
정보가 검색결과로 나타났고, DHT의 특징으로 인해 범위  
검색의 어려움이 있었다. 이를 해결하기 위해 우리들은  
DHT 기반의 P2P 시스템에 시멘틱 웹[7] 기술을 적용하

고, 범위 검색을 지원하기 위한 방안을 제시한다.

### 2. 관련연구

이 장에서는 본 논문의 기반 시스템인 MasicSquare의  
Naming & Key Location, 그리고 기본 구조를 알아본다.

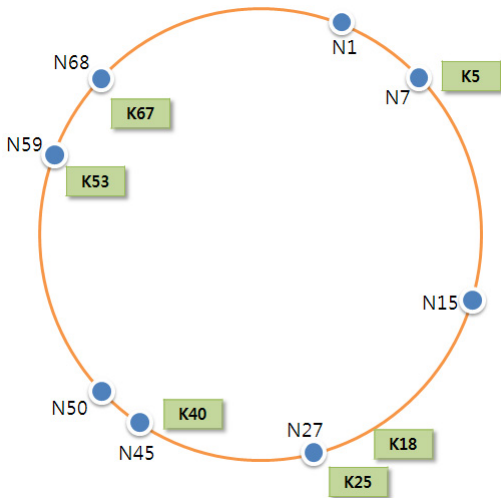
#### 2.1 MasicSquare의 Naming & Key Location

MasicSquare에서 피어를 구별하기 위해 고유의 ID 를  
160비트 SHA-1 해쉬 함수를 이용하여 만든다. 이 ID 는  
고정 IP 를 가지는 피어는 IP 를 해쉬하고, 동적 ID 를 가  
지는 피어는 사용자의 고유 ID 를 해쉬하여 만든다.

피어는 담당자(manager), 계승자(successor), 선행자  
(predecessor)로 구분된다. 담당자는 키의 값을 저장하는  
피어이며, 키의 값 k와 같거나 시계방향으로 바로 뒤에  
있는 피어이고, 계승자는 어떤 피어의 시계 방향으로 바로  
옆에 있는 피어이다. 그리고 선행자는 어떤 피어의 시계  
반대 방향으로 바로 옆에 있는 피어이다.

(a, b] 구간은 ID 공간에서 피어 a를 포함하지 않고, 시  
계 방향으로 뒤에 있는 피어 b를 포함하는 구간을 표기한  
다.

(그림1) 은 MagicSquare에 8개의 피어가 연결되어 있  
고, 6개의 키를 저장하고 있는 그림이다. 키 K25 와 K18  
의 담당자는 N27 이며, N27 에 K25 와 K18 이 저장된  
다. N27의 선행자는 N15 이고, N27의 계승자는 N45이다.



(그림 1) 6개의 키를 저장한 8개의 피어가 구성하는 MagicSquare 해쉬 공간

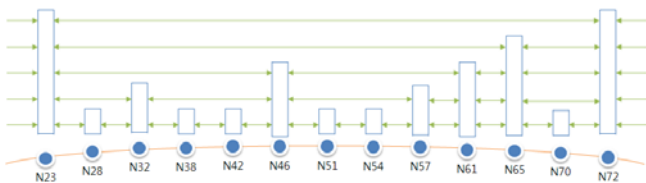
<표1> 파일기술을 위한 온톨로지

name	: 파일 이름
title	: 자료의 제목
type	: 파일의 종류로 문서, 영상, 이미지 등
category	: 자료의 분류. 영상의 장르, 영화 등
size	: 파일의 크기
format	: 파일의 형태
creator	: 파일을 만든 사람
modifier	: 마지막으로 파일을 변경한 사람
createdDate	: 파일을 생성된 날짜
modifiedDate	: 마지막으로 변경된 날짜
description	: 파일에 관한 설명
source	: 현재 파일이 파생된 자원에 대한 참조
relation	: 관련 파일들에 대한 참조
keyword	: 검색시 원하는 주제어
rights	: 파일에 대한 보유된 권리 정보

온톨로지는 특정 도메인에서 통용되는 의미 사전이다. 본문에서 제시하는 온톨로지는 특정 도메인인 “http://asadal.cs.pusan.ac.kr/~jwhan/off”에서 통용되는 온톨로지이다.

<표1>에서 Name은 파일의 이름이고, Title은 문서일 경우 제목을 의미한다. 파일명은 Tapestry에서 이미 고려된 상태이지만, Hash 값으로 찾아오기 때문에 루트노드가 관리하는 자료가 모두 동일한 검색어라고는 볼 수 없다. 그렇기 때문에 여기에선 파일이름이 명시되어야 한다. 그리고 Type은 자료의 형식이 무엇인지 나타내며, Category는 자료의 분류나 장르를 의미한다. 예를 들어 Title이 “전쟁과 평화”라는 자료는 Type이 “audio/mpeg”고 Category는 “멜로”이다. 이렇게 rdf를 이용하여 본 시스템 이용자는 자신이 원하는 자료를 보다 정확하게 찾을 수 있게 지원한다.

### 2.1 MasicSquare의 기본구조



(그림 2) MagicSquare의 구조 (전체 ID 공간 중 일부분)

MagicSquare에서는 각 피어들 간의 연결성을 (그림 2)와 같이 스킵리스트 모양으로 구성되며 라우팅 또한 스킵리스트를 따른다. 위 그림은 N23에서 N72까지 부분 구간의 모습을 그림으로 나타낸 것이다. N23과 N72의 성능이 가장 우수한 피어 집합에 포함되며, 라우팅 테이블의 최고 높이는 5라고 가정한다. 피어 사이의 연결 정보와 다른 피어의 정보는 해당 피어의 라우팅 테이블에 저장하며, 2\*(라우팅\_테이블\_높이) 개의 피어 정보를 관리한다. 여기에서 라우팅 테이블 높이는 피어의 자원 공유 신뢰도, 컴퓨팅 능력, 메시징 능력 등을 고려하여 결정되며 성능이 좋을수록 높은 라우팅 테이블 높이를 가지게 된다.

## 3. 시스템 설계

이 장에서는 MagicSquare에 정보 검색의 효율을 위해 RDF[8]를 접목하고, 범위 검색을 위해 제안하는 시스템을 살펴본다.

### 3.1 파일 기술을 위한 온톨로지

RDF 형식의 온톨로지는 이미 널리 사용되며, 그 대표적인 예로는 문헌을 기술하기 위해 정의된 더블링크어 메타데이터[9]가 있다. 우리가 제안하는 시스템에선 정보의 검색할 때의 모호성을 줄이기 위해 파일기술을 위한 온톨로지(off : Ontology For File-description)[10]를 사용한다.

### 3.2.1 범위 검색을 위한 시스템

범위를 검색하기 위해선 기본적으로 정보의 접근 형태가 선형구조를 지원해야 한다. 이는 특정 범위에 대한 접근을 효율적으로 지원하기 때문이다.

시스템 관리자는 특정 온톨로지에 범위 검색이 가능하게 설정하면 시스템은 그 온톨로지를 해쉬해 그 결과값으로 담당자 노드(피어)를 찾게 된다. 이렇게 찾은 노드는 그 온톨로지의 R\_Node 노드라 칭하며, 이 R\_Node 노드부터 범위는 시작이 된다.

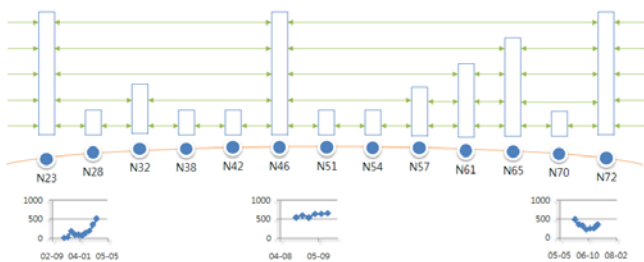
시스템의 최대 존재 가능한 노드의 개수를 N이라하고, 범위 정보가 가질 수 있는 최대값에서 범위 정보가 가질 수 있는 최소값을 뺀 값을 RV라고 하면 하나의 노드가 가질 수 있는 범위 정보값(NRV)은  $NRV = RV/N$ 이다.

각 노드는 자신의 범위에 해당하는 범위의 값을 관리한다. 그럼 시스템 전체는 R\_Node 노드부터 NRV 값만큼 해당하는 범위 정보를 담당하게 되어 범위 검색을 위한 순차적인 인덱스를 만들게 된다.

### 3.2.2 조밀 구간의 평활화 기법

검색어에 인기어가 있듯이 범위 또한 특정 구간에 많은 정보가 몰려 있을 수가 있다. 이는 결국 특정 구간을 담당하는 노드에 많은 질의가 몰리게 된다. 이를 위해 우리는 적응적 시스템을 제안한다.

본 시스템은 특정 구간에 정보가 밀집되는 것을 검사하기 위해 히스토그램을 이용한다. 이 히스토그램은 MagicSquare 상에서 라우팅 테이블 높이가 가장 높은 노드들이 관리를 하고 이 노드들을 S\_Node라 칭한다. 하나의 S\_Node는 자신과 다음 S\_Node 사이의 범위에 대한 히스토그램을 관리하며, 이 범위 사이에 있는 각 노드들이 관리하는 범위 정보의 개수가 평활화 함수( $c*ARN$ ) 보다 커질 때 평활화를 한다.



(그림 3) MagicSquare와 S\_Node 히스토그램

S\_Node가 관리하는 범위 안에 있는 전체 정보의 수를 TRV라고 하고, 이 노드가 관리하는 노드의 개수(최대 가능 노드 수가 아닌 실제 참여해 있는 노드의 수)를 RN이라 할 때, S\_Node가 관리하는 영역에서 평균 범위 정보의 개수  $ARN = TRV/RN$  이다. 이때 특정 노드가 관리하는 범위 정보 개수가  $c*ARN$  보다 크게 되면 평활화를 하여 각 노드가 관리하는 범위 정보 개수가 ARN이 되게 각 노드의 관리하는 범위 값을 조정하게 된다.

S\_Node가 자신이 관리 하는 범위 안에서 평활화를 하여 특정 구간 노드들에 범위 정보 값이 밀집되지 않게 유지 하듯, 시스템 전체의 평활화는 R\_Node 노드가 각 S\_Node에 대해 위와 같이 방법으로 관리함으로써 전체 시스템의 평활화가 이루어진다.

### 3.2.3 피어 결함 복구 방법

P2P 시스템에서의 피어가 시스템을 떠나는 경우는 크게 두 가지로 나눌 수 있다. 첫 번째는 종료 사실을 미리 알리는 정상적 이탈이고, 두 번째는 예고 없이 갑작스럽게 접속이 끊긴 비정상적 이탈이다. 이런 이탈에 대한 기본적인 복구 시나리오는 MasicSquare를 따른다.

정상적 이탈일 경우 자신과 연결된 모든 피어와 S\_Node에게 이 사실을 알리고 자신의 자원을 책임질 피어에게 지원을 복사시켜 시스템 상의 서비스가 계속 될 수 있게 한다.

비정상적 이탈일 경우 피어의 단절을 알아내고 스킵리

스트 연결을 재구성하는 추가의 과정이 필요하게 된다. MasicSquare에 참여한 피어들은 주기적으로 자신과 연결된 피어들과 메시지 (Living Message) 교환을 통해 단절 여부를 알아내고, 우회하는 경로를 통해 끊어진 경로를 다시 이어주거나, 같은 라우팅 테이블 높이를 가지는 피어에 대한 추가적인 링크를 가지고 있어 단절이 생길 경우 이 링크를 통해 다시 이어주게 된다. 그리고 S\_Node에게 변화된 부분을 알려주어 이를 시스템에 반영하게 된다. 이외에 S\_Node가 지역을 담당하기 때문에 S\_Node를 통해 직접적으로 링크를 수정 할 수 있는 방법도 있다.

범위 색인의 경우 같은 라우팅 테이블 높이를 가지는 피어들 사이에 중첩되게 범위 정보를 둬므로 피어가 비정상적으로 이탈할 때에도 범위 색인의 유실을 막아 지속적인 범위 검색을 가능하게 한다.

## 4. 결론 및 향후 연구 과제

P2P는 인터넷 등과 같은 자원 공유를 위한 시스템뿐만 아니라 센서 네트워크 등과 같은 그리드 환경에서 탐색 경로를 제안 한다. 이런 환경에서 자원탐색의 경로를 줄이고 시스템의 부하를 낮추는 것은 중요하다. 현재 DHT를 사용하는 시스템은 모두  $O(\log N)$ 의 성능을 보여준다. 하지만 이 시스템 중에는 물리적 위치를 고려하지 않았거나 고려하였다 하더라도 피어의 성능 차이를 고려하지 않았다. 그리고 DHT 환경의 특징으로 인해 범위 검색의 어려움이 있다. 우리는 본 논문에서 DHT 환경에서 범위검색을 용이하게 하기 위한 시스템을 제안하였고, 그 기반 시스템으로 피어의 물리적 위치와 피어의 성능을 고려한 MasicSquare를 이용하였다. 그리고 히스토그램을 이용하여 범위 검색을 위한 색인이 특정 지역이 밀집되는 것을 방지 하였다.

우리가 제안한 평활화를 위한 함수  $c*ARN$ 에서 관리자가 정하는 상수  $c$  값이 너무 작으면 빈번한 평활화 작업이 일어나고,  $c$  값이 너무 크면 범위 정보가 밀집되는 현상이 나타나는데 이를 적절히 조절할  $c$  값을 아직 제안하지 못 했고, 범위 값을 적절히 분산 시켰다 하더라도 특정 부분에 질의가 빈번할 수 있지만, 아직 이 부분은 고려하지 않았다. 이 문제들은 차후에 더 살펴볼 문제들이다.

### 참고문헌

[1] Napster, <http://www.napster.com>  
 [2] Sylvia Ratnasamy, Paul Francis, Mark Handley and Richard Karp, A Scalable Content-Addressable Network, ACM SIGCOMM'01, August, 2001  
 [3] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek and

Hari Balakrishnan, Chord: A Scalable Peer-to-peer  
Lookup Service

for Internet Applications, ACM SIGCOM'01, August,  
2001

[4] Antony Rowstron and Peter Druschel, Pastry:  
Scalable,

decentralized object location and routing for large-scale  
peerto-

peer systems, IFIP/ACM International Conference on  
Distributed Systems Platforms (Middleware 2001),  
November, 2001

[5] Ben Y. Zhao, John Kubiawicz and Anthony D.  
Joseph, Tapestry:

An Infrastructure for Fault-tolerant Wide-area Location  
and

Routing, Technical Report, UCB/CSD-01-1141,  
Computer Science

Division, University of California, Berkeley, April, 2001

[6] 손영성, 김강범, 한동윤, 김경석, MagicSquare를 이용  
한 애플리케이션 레벨 멀티캐스트. 2004 가을 학술발표논  
문집(III) 제31권 2호 pp.589~591 한국정보과학회  
(2004/10)

[7] <http://www.w3.org/DesignIssues/Semantic.html>.  
Semantic Web Road map

[8] <http://www.w3.org/RDF>. World-Wide Web  
Consortium: Resource Description Framework

[9] <http://www.dublincore.org/about/overview/>. Dublin  
Core Metadata Initiative

[10] 한종욱, 이승은, 한동윤, 김경석, 효율적인 P2P 파  
일 검색을 위한 RDF 파일 온톨로지 구조. 2006년도 한  
국정보과학회 가을 학술발표논문집 (D) 제33권 2호  
pp.637~641 한국정보과학회 (2006/10)