

# Structured P2P에서 트리 구조를 이용한 Range Query의 구현

박성진<sup>0</sup>, 권오훈, 김종  
포항공과대학교

{taiji76<sup>0</sup>, dolphin, jkim}@postech.ac.kr

## Implementing Range Queries in Structured P2P Systems using Tree-Structure

Sung-Jin Park<sup>0</sup>, O-Hoon Kwon, Jong Kim  
Dept. of Computer Science and Engineering,

Pohang University of Science and Technology (POSTECH)

### 요 약

Distributed Hash Table(DHT)를 기반으로 하는 structured P2P에서 복잡 질의를 통해 원하는 자료를 찾는 것은 쉽지 않다. 이와 같은 것들 중 하나가 범위 값을 이용하여 원하는 값을 찾는 range query이다. 본 논문에서는 full binary tree의 특징을 이용하여 structured P2P에서 range query를 수행할 수 있는 mapping 함수와, load balance를 지원하기 위한 방법을 제안한다. 또한 실험을 통해서 이 방법들이 어느 정도의 성능을 보여주는지도 살펴본다.

### 1. 서 론

P2P(Peer-to-Peer)란 해당 프로토콜에 참여하는 모든 peer들이 서버와 클라이언트의 역할을 동시에 수행하며, 전체 시스템에 대한 정보가 없어도 원하는 기능을 수행할 수 있는 구조를 의미한다. 이를 간단하게 구분하면 structured P2P와 unstructured P2P로 구분할 수 있다. Structured P2P의 경우 DHT를 기반으로 하여, 임의의 peer나 자료의 위치를 결정하여 찾아갈 수 있는 장점이 있는 프로토콜로 Chord, Pastry등과 같은 것들이 개발되어 있다. Unstructured P2P는 peer나 content가 임의로(random) 분포가 돼서 그 위치를 알 수 없는 구조를 의미하며, Napster, JXTA등과 같은 프로토콜이 있다.

P2P 컴퓨팅은 많은 장점들로 인해 데이터 베이스를 P2P로 적용하거나[5], Grid의 information service를 P2P를 이용해 구현하려는 것과 같은 다양한 분야에 적용하려는 노력이 이루어지고 있다. 이러한 적용 노력에도 불구하고, 많은 문제점들로 인해 실제 적용은 잘 되지 않고 있는데, 그 이유 중 하나가 다양한 질의를 지원하기가 힘들다는 것 때문이다. 이는 특히 structured P2P에서 DHT의 특성 때문에 더 힘들게 된다.

이러한 다양한 질의중 하나가 range query이다. 이는 말 그대로 찾기 원하는 범위를 질의했을 때, 그 범위에 해당하는 값들이 어떤 peer에 저장되어 있는가를 찾는 것을 의미한다. 이 range query의 경우 연속적인 값에 대한 질의를 의미하기 때문에 이산적인 값을 처리하는 DHT를 기반으로 하는 경우에 구현하기 힘들 뿐만 아니라, 표현을 한다 하더라도 원하는 값이 해당 peer에 없는 경우 어느 peer에게 질의를 전달하여 값을 찾아갈 수 있을지 결정하는 것이 쉽지가 않다. 따라서 본 논문에서는 DHT를 기반으로 하는 structured P2P에서 효율적인 range query를 구현할 수 있는 방법을 제시하고 이에 대한 성능 평가를 수행한다.

본 논문은 다음과 같이 구성된다. 2절에서는 연구 동기에 대해서 언급하고, 3절에서는 제안하는 매핑 함수에 대해서 설명한다. 4절에서는 실험결과에 대해 설명한 후에 5절에서는 결론을 내린다.

### 2. 연구 동기

Grid환경에서 사용자가 필요로 하는 자원을 찾아주는 기능을 하는 것이 information service이다. 그런데 Grid 환경에서는 많은 수의 컴퓨터가 자신의 자원을 제공하면서 Grid 환경에 참여하고 있고, 제공하는 자원의 종류도 매우 다양하다. 또한 제공한 자원들이 어떻게 사용되는지 살펴보기 위해서는 모니터링을 하는 프로그램도 필요한데, 이 또한 자원을 찾아주는 것을 기반으로 하여 모니터링 서비스를 제공하게 된다.

위와 같이 많은 연산이 몇 개의 information service에 집중되므로 인해 부하가 발생할 수 있게 된다. 이와 같은 문제점을 해결하기 위해서 참여하는 모든 컴퓨터가 P2P 프로토콜을 이용하여 information service를 수행하는 것에 대한 많은 연구가 수행되었다.[1, 2, 3]

그러나 기존 연구의 경우 단순히 range query를 구현하는 것에만 목적을 두었고, 전달되는 질의의 횟수나 찾아진 범위값의 형태, 즉 가장 좁은 범위의 결과나 넓은 범위의 결과를 찾아주는 것에 대한 고려는 없었다. 본 연구에서는 full binary tree의 특성을 이용하여 앞에서 언급한 문제점을 해결할 수 있는 매핑 함수를 제안한다.

### 3. 제안하는 range query 알고리즘

#### 3.1 시스템 모델

Range query를 구현하는 방법은 크게 두가지로 나눌

수 있다. 새로운 P2P 프로토콜을 디자인하는 것과 기존의 프로토콜을 수정하는 방법이 있다. 본 연구에서는 두 번째 방법을 사용하여 구현하는 것을 목표로 했다.

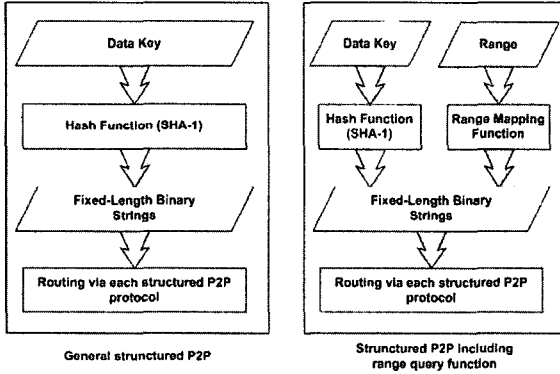


그림 1 시스템 구조

그림 1에서 왼쪽이 기존의 structured P2P의 구조이다. 제공하려는 자료의 키값을 SHA-1과 같은 해쉬 함수를 사용하여 고유의 고정된 길이의 값을 얻고, 이를 고유의 프로토콜을 이용하여 원하는 peer에게 전달하는 방법을 사용한다. 본 연구에서는 오른쪽에 있는 그림과 같이 keyword를 사용하는 자료는 그대로 해쉬 함수를 사용하고, 범위에 대한 질의만 본 연구에서 제안하는 매핑 함수를 이용하여 고정된 길이의 값을 얻게 된다. 이 얻어진 값을 이용하여 기존의 P2P 프로토콜을 이용하여 원하는 peer에게 질의를 전달함으로써 범위에 대한 질의를 수행하게 된다. 이와 같은 방법을 사용하기 때문에 기존의 structured P2P에서 range query를 수행할 수 있게 된다.

### 3.2 Full binary tree의 특징

앞부분에서 언급한 것과 같이 제안하는 알고리즘에서는 full binary tree의 특징을 이용한다. 임의의 범위가 어느 위치에 저장되는지는 실제로 트리 구조를 생성해서 알아내는 것이 아니라, full binary tree에 해당하는 간단한 공식을 이용하여 그 위치를 알아내는데 이에 사용되는 공식은 아래와 같다.

- Full binary tree의 높이 :  $\lceil \log_2(n+1) \rceil$
- 임의의 높이에서 가장 왼쪽에 있는 트리 노드의 번호 :  $2^{h-1} - 1$
- 현재 트리 노드에서 부모 노드의 번호 :  $(n-1)/2$

위의 공식에서 트리의 노드의 번호는 루트 노드를 0으로 하여 왼쪽의 노드부터 번호를 부여하는 것으로 가정한다.

### 3.3 문제 정의

기존 연구에서 해결하지 못한 문제점들을 해결하기 위하여 아래와 같은 특징을 가지는 매핑 함수를 필요로 한다.

- ① 기존에 존재하고 있는 structured P2P에 적용할 수 있어야 한다.
- ② 만약 P2P system에 찾으려는 range가 있다면 반드시 찾을 수 있어야 한다.
- ③ 가장 좁거나 넓은 범위의 값을 찾을 수 있어야 한다.
- ④ 범위 값을 찾음에 있어 upper bound가 있어야 한다.
- ⑤ 제공되는 범위 값을 분산시킬 수 있는 방법이 있어야 한다.

### 3.4 제안하는 매핑 함수

앞에서 언급한 것과 같이 트리를 이용하여 범위가 저장될 위치를 결정한다. 그림 2는 트리에서 한 노드가 의미하는 것을 나타낸 것이다. 범위가  $A \leq \text{memory} \leq B$ 일 때, 트리의 한 노드에 매핑 됐다고 가정하자. 이는 범위의 최소값인 A가 노드말에 명시돼 있는 1~50사이에 포함되고, 질의의 범위인 B-A가 옆의 range 51~100사이에 포함됨을 의미한다. 트리의 노드의 번호는 시스템 모델에서 보인 고정된 길이의 값이 되며, 이 노드 번호를 이용하여 peer의 위치를 찾아가게 된다.

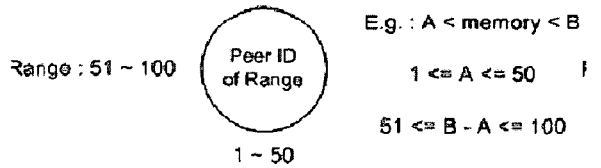


그림 2 노드 표현 방법

위와 같은 의미를 가진다는 것을 알고 그림3의 전체 구조를 통해 어떤 과정을 통해 저장될 peer의 위치가 결정되는지 확인해 보도록 하자. 우선 범위가 결정되는 과정을 나열하면 아래와 같은 순서를 통해 위치가 결정된다.

- ① 범위값이 저장될 위치를 찾기 위해서 매핑 함수를 호출한다.
- ② 범위가 트리에서 어느 깊이에 해당하는지 계산한다.
- ③ 계산된 깊이에서 가장 왼쪽에 있는 노드의 번호를 계산한다.
- ④ 계산된 노드 번호부터 오른쪽에 해당되는 노드를 순서대로 확인하면서 최소값이 해당 노드에 포함되는지 확인한다.
- ⑤ 확인된 노드의 번호가 매핑 함수에서 리턴하는 값이므로 이 값을 리턴한다.
- ⑥ 리턴된 값을 이용하여 기존의 P2P 프로토콜을 이용하여 원하는 peer를 찾아 라우팅을 수행한다.

위와 같은 과정을 이용하여 위치를 결정하게 된다. 그림3과 예를 통해서 그 과정을 살펴보도록 하자.

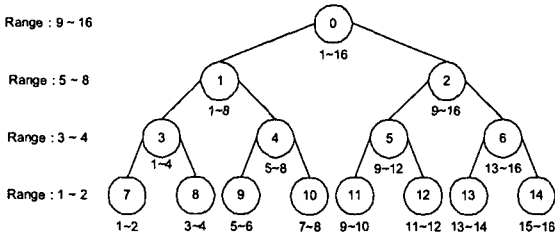


그림 3 전체 full binary tree구조

7 < memory < 9에 해당하는 범위값이 어느 peer에 저장되어 있는지 위의 과정을 이용하여 계산해 보도록 하자. 이 범위값이 매핑 함수의 인자로 전달되면, ②에서 처럼 트리에서 깊이를 구한다. 질의로 전달된 범위가 9 - 7 = 2이므로 깊이가 4가 된다. ③에 의해서 가장 왼쪽에 있는 노드의 번호를 구하면  $7(=2^{(4-1)}-1)$ 이 된다. ④에서와 같은 방법을 이용하여 7번 노드에서 오른쪽으로 이동하면서 범위의 최소값인 7이 어느 노드에 해당되는지 계산해보면 10번 노드에 포함되는 것을 알 수 있다. 이 값이 매핑 함수에서 리턴하는 값이 된다. 이 값을 이용하여 기존의 P2P 프로토콜을 이용하여 찾아가게 된다.

이와 같은 트리 구조를 사용함으로써 원하는 값이 해당 peer에 없는 경우 그 부모 노드에 해당하는 peer에게 질의를 보냄으로써 불필요한 질의의 전달을 막을 수 있다. 또한 제안하는 방법이 범위값을 우선시 해서 계산하게 되는데 이 때문에 가장 좁거나 넓은 값을 찾아주는 것이 가능하게 된다. 또한 어느 위치에 값이 있다는 것을 알고 질의를 전달하기 때문에 upper bound안에 원하는 값을 찾는 것이 가능하게 된다. 그러므로 문제 정의에서 명시한 내용들을 만족하는 range query를 수행할 수 있는 매핑 함수를 구현했음을 알 수 있다.

3.5 성능 향상을 위한 방법

일반적으로 여러개의 memory와 같은 하나의 값이 아닌, CPU, 저장 크기와 같은 여러 종류의 값을 저장하려고 하는 경우에, 값의 분포가 비슷하게 된다. 예를 들면 대부분의 사람들이 자신이 제공하려는 것은 적은 양만 제공하려는 성향을 가지기 때문에 몇 개의 peer에 값이 집중돼서 저장되는 경우가 발생하게 된다.

위와 같은 것을 막기 위해서 원점, 즉 각 요소들이 시작되는 점을 전부 다르게 하면 값을 분산시킬 수 있다. 본 연구에서는 이를 위해 요소들의 이름을 해쉬 함수에 넣어 그 값만큼 원점을 증가시키는 방법을 사용하였으며, 실험 결과의 Diff가 이 방법을 사용한 것을 의미하고 Same은 0을 원점으로 해서 실험한 것을 의미한다.

4. 실험 결과

제안하는 알고리즘을 structured P2P중 하나인 Chord[4]를 이용하여 구현해 보았다. 이전에 설명한 것과 같이 매핑 함수에서 나온 위치값을 기존의 프로토콜

을 그대로 이용하여 질의를 목적지 peer에게 전달하기 때문에 당연히 적용되는 프로토콜의 라우팅 길이와 같은 결과를 나타낸다. 그러므로 실험에서는 Chord와 같이  $O(\log_2 n)$ 만큼 질의가 이동을 하면 원하는 peer에 도착하게 된다.

		100개의 질의			
		평균	최대 (load)	편차 (load)	편차 (전달)
Uniform	Same	3.35	3.94	0.29	10.01
	Diff	3.28	1.43	0.14	4.82

표 1. 실험 결과

Load balance를 수행하는 방법을 사용한 경우 아래의 표와 같은 결과를 보여준다. 최대의 load를 갖게 되는 peer의 경우 적용 전 약 4%에서 적용 후 약 1.2% 정도로 약 3%정도 줄어드는 것을 볼 수 있고, 전체 peer에 분산된 load의 경우 표준편차가 0.3에서 0.14로 약 50% 정도로 줄어들었음을 확인할 수 있었다.

5. 결론

본 연구는 structured P2P에서 DHT에서 사용하는 해쉬함수 대신 범위 값의 위치를 결정하기 위한 매핑 함수를 이용하여 range query를 구현하는 방법과 load balance를 위한 알고리즘을 제안하였다. 또한 이를 Chord를 이용하여 구현하였고, 제안하는 알고리즘의 효율성에 대해 실험하였다.

본 연구에서는 여러 종류의 범위값이 질의로 들어온 경우에 이를 하나 씩 나누어서 질의를 수행한 후, 나중에 원하는 값을 찾아내는 것으로 가정하였다. 향후 연구에서는 이를 동시에 처리할 수 있는 방법에 대하여 연구할 것이다.

참고문헌

- [1] Kothari, D. Agrawal, A. Gupta, S. Suri, "Range Addressable Network: A P2P Cache Architecture for Data Ranges", 3rd International Conference on Peer-to-Peer Computing (P2P'03)
- [2] A. Iamnitchi and I. Foster, "On Fully Decentralized Resource Discovery in Grid Environments", International Workshop on Grid Computing, Denver, CO, November 2001
- [3] Artur Andrzejak, Zhichen Xu, "Scalable, Efficient Range Queries for Grid Information Services", 2nd International Conference on Peer-to-Peer Computing (P2P'02).
- [4] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications", IEEE/ACM Transactions on Networking.
- [5] S. Gribble, A. Halevy, Z. Ives, M. Rodrig, and D. Suci, "What can peer-to-peer do for database and vice versa?", In 4th International Workshop on the Web and Databases (WebDB '01)