

C-Chord: 분류 검색을 위한 DHT 기반의 P2P 시스템†

김상영⁰ 박재현
중앙대학교 컴퓨터공학과
sykim@hsc.cau.ac.kr⁰, hyunie@cau.ac.kr

C-Chord: The DHT-based P2P System for the Categorized Search

Sam-Young Kim⁰ Jae-Hyun Park
Dept. of Computer Science and Engineering, Chung-Ang University

요 약

P2P 시스템들의 주요한 이슈는 효율적인 검색 기법이 있다. 기존의 DHT(distributed hash table) 방식의 P2P 시스템들은 단순 쿼리에 이용되는 단일 키워드 일치만을 통한 검색 기법에만 머물렀다. 본 논문에서는 이러한 단순 쿼리 기반의 검색 기법을 개선하기 위해 DHT 기반의 도큐먼트 라우팅 모델에서 다중 링 토폴로지를 이용한 분류 탐색 기법을 제안한다. 제안된 기법은 Chord[1] 프로토콜을 기반으로 구현하였으며, 분류 검색의 지원 이후에도 로드 밸런싱에 있어 기존 시스템과 유사한 성능을 보임을 확인한다.

1. 서 론

인터넷의 폭발적인 사용자 증가와 함께 정보공유의 개인화에 따라 P2P(Peer-to-Peer) 네트워크를 이용한 대규모 분산 시스템이 새로운 이슈로 성장하게 되었다. 이러한 기존의 P2P Overlay Network에서는 키워드의 완전 일치 또는 부분 일치를 통한 정보의 검색이 이루어졌다. 하지만, 현실에는 다양한 분류 가능한 정보들이 존재하고, 실제로 사용자들은 분류된 범위 내에서의 정보 검색을 요구한다. 실제로 Yahoo!와 같은 주요 서버-클라이언트 시스템들이, 분류 검색을 지원하는 것을 볼 수 있다. 대표적인 비구조적 P2P 시스템인 Gnutella[2] 등의 브로드캐스팅 기법에서는 일단 모든 Peer에게 복잡한 질의를 전달이 가능하므로, 범위 검색, 분류 검색 등의 복잡한 쿼리를 수행할 수 있지만, 브로드캐스팅 기법 자체의 과도한 트래픽 발생이라는 문제점을 안고 있다. 이를 보완하기 위한 브로드캐스팅 범위의 제한 등으로 인해 검색에 대한 응답의 불규칙성 및 검색 결과의 완전성에도 문제를 갖게 된다. 이에 반하여 Chord 등의 도큐먼트 기반의 P2P 시스템에서는 DHT 기법을 사용하기 때문에 검색 완전성에서는 높은 성능을 보이지만, 키워드 완전 일치를 제외한 복잡한 쿼리의 경우에는 수행에 한계가 따른다. 그러므로, DHT 기반의 P2P 검색 시스템을 위한 Range 검색, Category 검색 등의 복잡한 쿼리를 지원하는 새로운 기법들이 요구 된다. 본 논문에서는 검색 완전성이 높은 DHT기반의 구조적 P2P Overlay Network분야에서 Web Services 또는 Grid

를 위한 Resource Discovery 분야에 적합한 분류 검색 시스템을 제안한다.

2. 관련 연구

순수 P2P 네트워크의 대표적인 DHT기반 프로토콜인 Chord 시스템과 대표적인 Service Discovery 기술인 UDDI 대하여 살펴본다.

2.1. Chord [1]

Chord는 대표적인 DHT 기반의 도큐먼트 라우팅 모델 방식의 프로토콜로 SHA-1같은 해쉬 함수를 사용하여 원형의 m-bit 식별자 공간에 각 노드와 키를 할당한다. 일반적으로 object의 위치 정보가 어떤 노드에 보관 될 지가 object 이름의 해쉬 값에 따라 (K, V)와 같은 쌍으로 정해지고, 해쉬의 성질에 따라 각 위치 정보들은 overlay network 상에서 균일한 분포로 pseudo-random한 위치에 유지될 것을 기대할 수 있다. 그러므로 해쉬의 성질만으로 object의 위치 관리에 있어서 로드 밸런싱 효과를 얻을 수 있다.

키(key)를 해쉬한 값을 식별자 공간으로 대응시킬(mapping) 때는 consistent hashing과 같이 원형 식별자 공간에 해쉬 된 키 값이 같거나 더 큰 노드 식별자를 갖는 첫 번째 노드에 저장한다(successor). 또한, Finger table이라는 라우팅 테이블을 두어 키를 삽입하거나 키를 가진 노드를 찾기 위한 lookup 등의 메시지를 해당 노드로 전달하는데 이용하며 유지 정보를 최소화하기 위해 원형 식별자 공간에서 노

† 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았습니다.

드의 위치(p)를 기준으로 하여 시계 방향으로 거리를 지수적으로 증가시켜 가면서 $p+2^n$ 에 해당하는 식별자 값을 가진 노드들에 대한 정보만 유지한다. 또한 오버레이 네트워크의 안정을 위해 이전 노드(predecessor)에 대한 정보를 유지하고, successor list를 유지함으로써 node failure시의 복구 문제에 대하여 대비 하고 있다.

2.2. Services Discovery

UDDI는 Web Services를 위한 Resource Discovery System의 대표적인 예라고 할 수 있으며 웹 서비스를 이용하고자 하는 사용자는 UDDI를 통하여 키워드 및 카테고리를 지정하여 원하는 서비스를 검색 할 수 있다. 이러한 카테고리 분류를 위해 대부분의 UBR(Universal Business Registry)들이 UNSPSC[3], NAICS등의 분류 표준을 사용하고 있다. 또한, UDDI Registry들은 일정 주기로 서로의 업데이트된 정보를 복제하기 때문에 동기화 이전에는 일치하지 않는 정보를 가지고 있을 수도 있다. 이러한 중앙집중형태의 UDDI Registry가 가지는 동기화 문제를 해결하기 위해서 P2P 기법을 이용한 Registry 구성 및 복잡한 쿼리 수행에 대한 연구가 진행되고 있다[4].

DHT기반의 P2P 시스템이 가지는 장점 중 하나인 식별자를 통한 로드 밸런싱을 유지하려면 분류 검색을 지원하는 동시에 분류된 object들의 key 분포가 식별자 공간 전체에 균일하게 분포 되어야 한다. 기존 Chord시스템의 Ring Topology를 그대로 이용하여 분류 검색을 수행하는 경우, 분류를 해쉬한 식별자에 해당 분류의 object 정보들을 모두 저장하게 되어, 인기 있는(hot spot) 분류의 경우 특정 노드에 다량의 키가 모이게 되고, 결국 다수의 lookup이 몇몇 인기 있는 노드에 집중되어 트래픽 불균형과 높은 부하로 인한 lookup 실패율의 증가를 초래하게 된다.

이를 해결하기 위해 본 논문에서는 [그림 1] (a)와 같이 링 상의 일정 범위에 계층적인 범주를 정하고 해당하는 계층의 범위 내에 object 정보들을 위치시킨다. 분류에 따른 key의 위치를 정하는 방법은 수식 $K_{location, level=a}$ 와 같다. a 는 object에 지정된 카테고리의 Level을 나타내며, 각 Level의 Range는 UNSPSC 등의 사용되는 분류 표준을 고려하여 [그림 1] (a)와 같이 일정 범위로 결정한다. (단, 카테고리가 지정되지 않은 object를 위한 $Level_0$ 의 범위는 링 전체이다.)

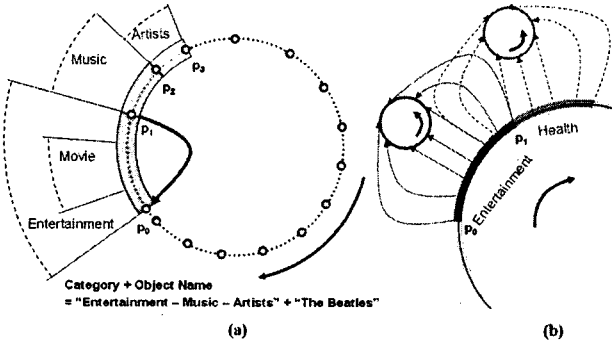


그림 1. (a) 원형 식별자 공간에서의 분류
(b) Sub Ring Topology 구성

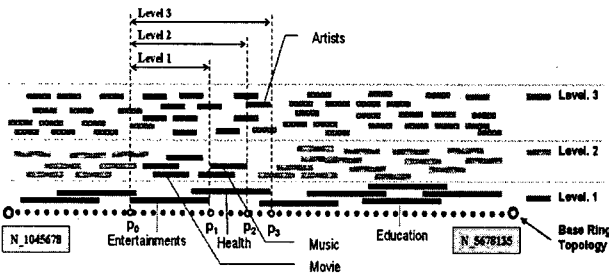


그림 2. Multiple Ring Topology

3. C-Chord 시스템

본 장에서는 DHT 기반의 도큐먼트 라우팅 모델인 Chord 시스템에서 다중 해싱 기법을 이용하여 분류 검색을 지원하는 방법을 제안 한다.

3.1. Key Location

$$K_{location, level=a} = \left(\sum_{i=1}^a hash(categoryId_i) \bmod (Range\ of\ Level_{i-1}) \right) + ((hash(objectId) \bmod (Range\ of\ Level_a)))$$

각 Level의 시작 위치는 상위 레벨의 범주 내에 존재하며, 시작 위치를 기준으로 지정된 Range만큼을 현 Level의 범주로 한다. 또한 이 범주 내에 하위 레벨의 시작 위치는 $hash(categoryId)$ 를 통해 결정하며, object의 key가 위치하는 경우에는 $hash(objectId)$ 를 통해서 결정한다.

예를 들어 $2^6(64)$ 개의 식별자 공간을 가지는 링 토폴로지에서 각 레벨의 range가 각각 “16, 8, 4”라고 할 때 카테고리가 “Entertainment-Music-Artists”인 “The Beatles”라는 object의 위치는 다음과 같다.

$$K_{location, level=3} = (hash(“Entertainment”) \bmod 64) + (hash(“Music”) \bmod 16) + (hash(“Artists”) \bmod 8) + (hash(“The Beatles”) \bmod 4)$$

최상위 레벨인 “Entertainment”의 모든 하위 구성원들이 위치할 수 있는 범주는 [그림 2]의 Level 3(p_0-p_3)가 되며, “The Beatles”는 p_2-p_3 사이에 위치하게 된다.

3.2. 분류 검색 기법

분류 검색을 위해 제안된 링 토폴로지에서는 세 가지 검색 방법을 지원한다. 첫째, 기본적인 링 토폴로지 상에 분포된 object를 이용한 키워드 검색. 둘째, 정해진 분류 내에서의 키워드 검색. 셋째, 정해진 분류내의 모든 object에 대한 Sequential 검색을 지원한다. Sequential 검색을 지원하기 위해서는 [그림 1] (b)와 같이 한 분류 범위 내 시작 노드

(p_0)와 마지막 노드(p_1)간에 링크를 유지하여 base ring내에 계층적인 링 토폴로지 구성이 가능하다. 서브 링에서의 Sequential 검색은 finger table의 ($p+2^0$)에 해당하는 식별자 위치를 이용한다. successor list를 통한 분류 내 순차 검색은 범위내의 첫 노드와 마지막 노드의 query전달 시간의 차이로 인해 마지막 노드는 첫 노드에 비하여 긴 응답시간 및 Path Length를 가질 것이다. 좀 더 효율적인 응답시간의 보장을 위하여 predecessor를 이용한 병렬 검색을 제안하며, 해당 레벨의 링에서 임의의 최초 접근 노드를 선정하고, 이를 기준으로 successor와 predecessor를 이용하여 양방향 Sequential 검색을 동시에 수행한다. 이 기법을 통하여 전체 노드의 평균 응답시간 감소를 기대할 수 있다.

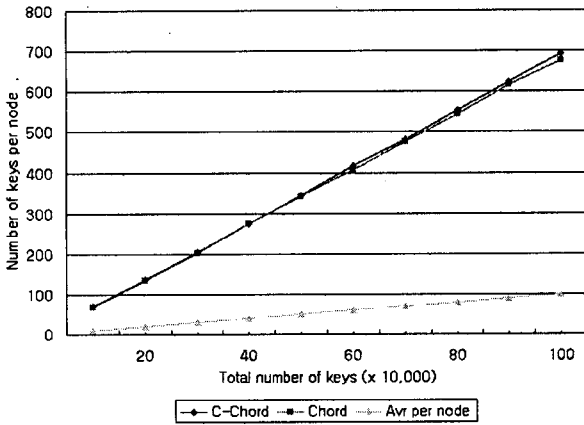


그림 3. Chord vs C-Chord의 Load Balance

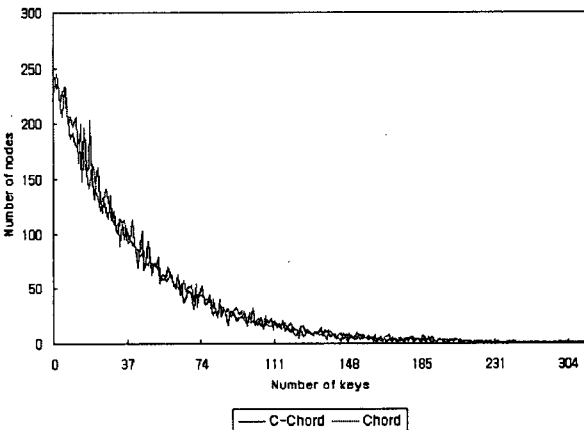


그림 4. C-Chord의 보유 key수에 따른 node의 빈도

4. 실험 및 결과

분류 검색을 지원하는 C-Chord와 지원하지 않는 Chord 시스템의 load-balancing 효과를 비교하기 위하여 수정된 Chord Simulator를 사용한다.

시뮬레이션은 10000개의 노드를 기준으로 10^5 개에서 10^6 개까지 10^5 개씩 object의 수를 증가시키면서 노드 당 보유 키 수를 확인한다. C-Chord의 경우 레벨 별 범위에 따라 실험

결과에 영향을 줄 수 있으므로 최대 Level을 3으로 제한하고 범위를 달리하며, 두 대상 모두 참여 노드의 id의 분포에 따라 Load Balancing에 영향을 받으므로 seed를 달리하여 여러 번 실험 후 평균값을 비교한다.

[그림 3]의 결과는 전체 키의 1~99%에 해당하는 키의 분포를 비교한 것이다. 두 시스템 모두 전체 keys수의 증가에 따라서 노드 당 최대 보유 keys도 함께 증가한다. 이는 분류 검색을 지원하지 않는 기존 Chord 시스템과도 유사한 부하 균등 효과를 보임을 나타낸다. 하지만, 전체 keys 수가 증가함에 따라 key를 링 전체를 대상으로 균등한 배치를 수행하는 Chord에 비하여 범위의 제약을 통해 분류를 지원하는 C-Chord간에 약간의 차이를 보임을 확인할 수 있다.

[그림 4]는 4×10^5 개의 보유 keys수에 따른 node의 분포로 두 기법 모두 유사한 지수적인 분포를 보임을 확인한다. 이러한 분포는 키가 집중된 몇몇 hot spot 분류에 동시에 Lookup이 집중되는 경우 순차 분류 검색의 첫 접근 노드에 부하가 집중될 수도 있으나 sequential 검색이 가능하므로 서브 링 상에서 순차 검색의 시작 위치를 임의로 선정이 가능하며, 이를 통해 양방향으로 검색을 수행하므로, 한 노드에 다수의 동시 접근으로 인한 부하를 분산시킬 수 있을 뿐만 아니라 검색의 응답 시간 및 평균 path length도 감소가 가능하다.

5. 결론 및 향후 과제

본 논문은 DHT 기반의 Ring Topology를 이용한 P2P overlay network에서 분류 검색을 제공하기 위하여 다중 해싱을 이용한 계층적 링 토폴로지 구성을 제안한다. 또한 이를 기반으로 기본적인 완전 일치 키워드 검색뿐만 아니라, 분류별 키워드 검색, 분류 별 Sequential 검색, 그리고, 병렬 검색이 가능하다.

시뮬레이션을 통하여 분류 검색을 지원하는 C-Chord 시스템이 분류검색을 지원하지 않는 Chord 시스템의 부하 균등 효과와 비교하여 유사함을 확인 하였다.

향후 연구 과제로는 전체 keys의 증가에 따른 load-balancing 효과를 개선하고, 분류 검색 시의 응답 시간 및 path length 감소 방안에 대하여 연구하고자 한다.

6. 참고 문헌

- [1] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," in Proceedings of ACM SIGCOMM, San Diego, CA, 2001, pp. 149-160.
- [2] Gnutella, <http://gnutella.wego.com/>
- [3] UNSPSC, <http://www.unspsc.org/>
- [4] Cristina Schmidt, Manish Parashar, "A peer-to-peer Approach to Web Service Discovery," World Wide Web: Internet and Web Information Systems, 7, 2004, pp.221-229