

JXTA 기반의 XPath 질의 처리 시스템

이지훈*, 손기락
한국의국어대학교 컴퓨터 및 정보통신공학과
e-mail : ainer@lycos.co.kr

An XPath Query Processing System based on JXTA

Ji-Hun Lee, Kirack Sohn
Dept. of Computer and Information Telecommunication Engineering,
Han-Kuk University of Foreign Studies

요 약

최근 MP3 저작권에 대한 찬반 논란이 계속 되면서, 기존의 클라이언트-서버가 아닌 Peer-To-Peer(P2P) 기반의 데이터 검색 시스템에 관한 여러 연구가 활발히 진행되고 있다. 하지만, 검색 방법에 있어서는 단순히 파일이름을 이용한 데이터 검색 방법이 주류를 이루고 있는데 본 논문에서는 검색을 위한 문서 정보를 XML 형태로 저장하고, 구조적인 검색을 용이하게 하기 위해 DOM(Document Object Model)을 이용하여 XML 문서를 트리 형태로 구성한 후 P2P 네트워킹 플랫폼인 JXTA를 기반으로 XPath 질의를 수행함으로써 P2P 네트워크 기반에서 구조적 검색이 가능한 시스템을 구현하였다.

1. 서 론

오늘날의 정보 시스템 환경은 인터넷의 무한한 발전과 함께 더욱 최적화된 환경으로 발전하고 있다. 네트워크를 통한 정보의 통신량이 증가하면서 기존 HTML 태그의 한계와 SGML 문법의 복잡성 등을 개선한 XML[1]이 새로운 문서 표현의 표준으로 각광 받고 있다. 문서의 의미 정보를 태그로 정의하여 사용할 수 있는 XML은 문서의 체계적인 관리 및 빠른 검색을 용이하게 한다. 하지만, 문서의 세부적인 검색에 있어서는 XML 문서의 구조 정보를 알지 못하면 문서 안에서의 특정 엘리먼트를 찾는 질의를 수행할 수 없다.

그래서, W3C에서는 XPath(XML Path Language)[2]를 제안하였다. XPath의 가장 큰 특징은 트리 구조의 XML 문서 탐색을 위해서 경로 정보를 사용하는 것이다. 인터넷 상에 존재하는 클라이언트-서버 모델 기반의 검색 시스템은 이러한 XML의 장점을 이용하여 수 많은 정보를 효율적으로 저장하고 관리할 수 있지만, 검색을 위한 서버의 성능이 시스템 전체 성능에 영향을 미치게 된다. 이러한 클라이언트-서버의 단점을 보완하기 위해서 P2P 네트워크 모델이 새로운 해결방법으로 제시되었다. 본 논문에서 적용된

JXTA 프레임워크[3] 또한 P2P를 지원하는 여러 플랫폼 중 하나이다.

본 논문에서는 최적화된 P2P 환경을 제공하는 JXTA를 기반으로 P2P 네트워크 상에서의 검색 속도 향상을 위해 Peer간의 정보를 공유하고 XPath 질의를 모든 Peer에 하지 않고 후보 Peer 리스트를 산출하여 질의함으로써, 효과적인 XML 문서검색 시스템을 설계하고 구현한다.

2. 관련연구

P2P 기반의 다양한 검색 엔진을 구현하는데 있어서 검색 방법의 효율성 보다는 검색 속도에 더 많은 비중을 두고 있다. 현재 상용화 되어 있는 그누텔라(Gnutella)나 냅스터(Napster)등과 같은 시스템들은 검색의 속도를 향상시키기 위해서 텍스트를 이용한 키워드 비교 검색 방법을 사용하고 있다[4][5].

P2P 검색 엔진은 연결된 네트워크의 형태에 따라 크게 두 가지로 구분할 수 있다.

2.1. 하이브리드 P2P 모델

클라이언트-서버 모델과 P2P 모델의 혼합 네트워크

구조로서, 서버가 존재 하지만 서버의 역할을 최대한 줄이고 Peer들은 IP를 이용하여 서로 통신하는 방식이다. 서버가 각 Peer의 정보를 관리하므로 다양한 검색 방법을 적용할 수 있지만 서버의 성능이 전체 시스템에 영향을 미치게 되고, Peer의 개수가 증가할수록 시스템의 성능이 저하된다는 단점이 있다.

2.2. 순수 P2P 모델

서버의 개입 없이 순수하게 Peer끼리 서로의 정보를 주고받는 형태의 네트워크 모델로서, 서로 연결된 Peer가 클라이언트 뿐만 아니라 서버의 기능을 병행하므로 Peer간의 직접적인 검색이 용이하고 수 많은 Peer들이 하나의 네트워크에 접속할 수 있다는 장점이 있다. 하지만, P2P 네트워크 구조의 특성상 검색을 위한 Peer 탐색 시간을 줄이기 위해서 키워드 비교 검색과 같은 비교적 간단한 검색방법을 사용해야 하는 한계가 있다.

2.3. P2P 기반의 검색 엔진

현재 서비스 하고 있는 P2P 기반의 검색 엔진들은 서버에 의존하지 않고 Peer끼리 검색할 수 있다는 장점이 있지만, 키워드 비교 검색의 단점으로 인하여 검색하려는 정보와 상이하게 다른 결과가 반환될 수 있을 뿐만 아니라 Peer 전체를 탐색하므로 수행 속도가 현저히 감소된다는 단점이 있다. 그래서 P2P 기반에서의 검색 엔진들은 더욱 효율적인 검색방법의 필요성이 대두되었다.

3. XPath 질의 처리 시스템 설계

XPath 질의 처리 시스템은 P2P 방식의 장점을 이용하여 네트워크에 연결된 각각의 PC를 하나의 Peer로 간주하고, 자신의 인접 Peer의 정보를 서로 공유함으로써 서버없이 모든 사용자 간에 정보를 빠르고 정확하게 검색할 수 있도록 설계된 시스템이다.

임의의 Peer 그룹에 존재하는 서로 다른 Peer들은 각각이 소유하고 있는 XML 문서 정보를 자신의 인접 Peer에 수출하고, 정보를 받은 Peer는 XPath 질의 처리를 위해서 XML 문서의 정보를 DOM 인터페이스를 [6] 이용하여 트리 형태로 구성한다. 또한 XPath 질의를 수행하기 전에 키워드 질의를 먼저 수행해서 후보 Peer들을 선정하고 후보 Peer에게만 XPath 질의를 수행함으로써 효율적인 검색 시스템을 설계한다.

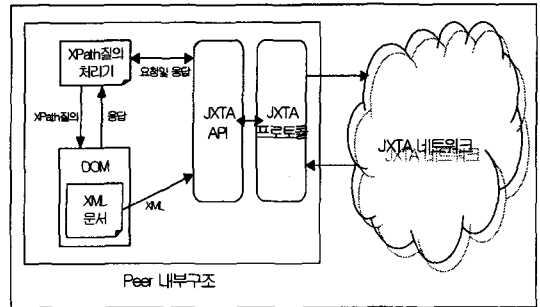
3.1. 시스템 구성

본 XPath 질의 시스템은 <그림 1>과 같이 XPath 질의에서 키워드를 추출하고 Peer 리스트를 조합하는 XPath 질의 처리기 부분과 다른 Peer와 정보를 공유하고 통신할 수 있도록 지원하는 JXTA[7] 함수 부분으로 구성된다.

Peer간의 모든 통신은 JXTA 프로토콜을 기반으로 통신하며, 각 Peer들은 JXTA에서 제공하는 함수를 이용하여 자신의 Peer 정보와 저장하고 있는 XML 문서의 정보, 그리고 인접 Peer에 대한 정보등을 서로

교환한다.

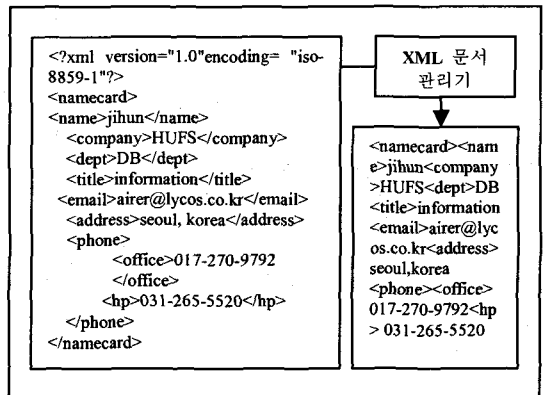
각 Peer들은 XML 문서를 저장하고, 자신이 가지고 있는 XML 문서에 대한 정보를 인접 Peer에 수출하기 위해서 <그림 3>과 같이 XML 문서에서 엘리먼트와 데이터만 추출한다.



<그림 1> 시스템의 전체 구조

```
<?xml version="1.0" encoding="iso-8859-1"?>
<namecard>
  <name eng="system">jihun</name>
  <company>HUFS</company>
  <major>DB</major>
  <email>airer@hufs.ac.kr</email>
  <interest>BasketBall</interest>
  <address>Seoul Korea</address>
  <phone>
    <office>031-332-4456</office>
    <hp>017-270-9792</hp>
  </phone>
</namecard>
```

<그림 2> namecard 문서의 실제 문서의 예



<그림 3> XML 문서에서 엘리먼트 및 데이터 추출

P2P 네트워크 기반 검색엔진의 단점 중 하나인 Peer 개수 증가에 따라 패킷 전송량이 증가하는 문제를 해결하기 위해서 모든 Peer에 XPath 질의를 수행하지 않고 응답할 수 있는 후보 Peer를 선정하여 해당 Peer들에게만 Xpath 질의를 수행한다. 그래서 XPath 질의를 수행하기 전에 <그림 4>와 같이 엘리먼트와 데이터를 키워드로 추출한다[8]. 하나의 JXTA 네트

워크에 연결된 모든 Peer들은 인접 Peer의 XML 문서에 대한 정보를 서로 공유하고 있으므로 인접 Peer에 키워드 검색 질의를 수행함으로써 각 키워드를 저장하고 있는 Peer 리스트를 얻을 수 있다.

XPath 질의	엘리먼트 및 데이터 추출
/namecard/name/title	namecard, name, title
/namecard/hp	namecard, hp
/phone[/office \$and\$ /hp]	phone, office, hp
/namecard[(\$not\$/office \$and\$ /hp]	namecard, office, hp

<그림 4> XPath 질의 분석

3.2. XPath 질의 처리

XPath 질의에서 추출한 각각의 엘리먼트와 데이터를 포함하고 있는 Peer 리스트들을 받으면 각 리스트들은 XPath 질의에서의 기본적인 연산자에 따라 다르게 조합된다. 기본적인 연산자의 우선 순위는 본 논문에서 정의한 BNF에 따라 조합되며, 나머지 연산자의 조합은 XPath의 BNF를 따른다.

XPath에서는 효율적인 트리 탐색을 위해서 “and”, “or”, “not” 등과 같은 Boolean 연산자와 함께 Filter와 같은 다양한 질의를 수행할 수 있는 함수를 제공한다.

예를 들어 XPath 질의를 분석할 때, “and” 연산자는 앞뒤 요소의 두개 Peer 리스트에 모두 존재 하는 Peer를 선택한다. 또한 “or” 연산자는 위의 경우 와 마찬가지로 앞뒤 요소의 Peer 리스트에 존재하는 모든 Peer를 선택한다.

마지막으로 “not” 연산자는 뒤 요소의 Peer 리스트에 존재하는 Peer를 제외한 나머지 Peer 리스트를 선택한다.

```
void Union() :// UNION 쿼리 수행
{ Intersection( (<UnionOp> Intersection))* }
[ 'OR' 연산자는 RightPeerList(Rlist) 와 LeftPeerList(Llist)를 OR해서 Llist를 push한다. ]

void Intersection() :// INTERSECT 쿼리 수행
{ Comparison( (<INTERSECT> Comparison))* }
[ 'AND' 연산자는 Llist와 Rlist를 AND해서 Llist를 push한다. ]

void Comparison() :
{ Path() | (<ANY> | <ALL>)? LValue() <ComparisonOp> RValue() }

void RValue() :
{ Path() | <NUMBER> | <TEXT> }

void Path() :
{ AbsolutePath() }

void AbsolutePath() :// ABSOLUTE 쿼리 수행
{ <SPATH> (RelativePath())?
  | <DPATH> RelativePath() }
[ AbsolutePath는 Rlist만을 선택하기 때문에 Rlist를 pop해서 SinglePath("/") 혹은 DoublePath("//")를 확인하고 push한다. ]

void RelativePath() :// RELATIVE 쿼리 수행
{ Subscript( ((<SPATH>|<DPATH>)RelativePath())? ) }
[ RelativePath는 Llist와 Rlist를 선택하므로 Rlist와 Llist를
```

각각 pop 해서 SinglePathP("/") 혹은 DoublePath("//")를 확인하고 Rlist에 Path 정보를 저장하고 Llist를 push한다.

```
void Subscript() :
{ Grouping( ("[" Subquery() "]" ) * ) }
[ XPath 쿼리에서 Filter 연산자("(")를 만나면 Filter안에 있는 쿼리를 하나의 Sub쿼리로 묶고 Sub쿼리의 Llist를 push한다. ]
```

```
void Negation() :
{ (<NOT>)? Union() }
[ 'NOT' 연산자의 Rlist를 제외한 나머지 이므로 Rlist를 pop한다. ]
```

<그림 5> XPath 질의 처리 시스템의 BNF

Q1. /name/phone/office

phone과 office는 상대패스('/')로 연결되어 있으므로 RelativePath()와 같이 Llist인 phone과 Rlist인 office를 질의해서 선택된 Peer 리스트를 저장하고 name을 질의한 Peer 리스트와 조합한다.

Q2. /phone[/office \$and\$ /hp]

/office와 /hp는 “and” 연산자로 연결되어 있으므로, Intersection()과 같이 office의 질의에서 선택된 Peer 리스트와 hp의 질의에서 선택된 Peer 리스트를 “and” 하고, Filter안에 존재하므로 Subscript()에서와 같이 하나의 Sub 쿼리로 만들고 phone을 질의한 Peer 리스트와 조합한다.

Q3. /namecard[(\$not\$/office \$and\$ /hp]

“not” 연산자는 Negation()과 같이 “not” 연산자의 Rlist인 office를 질의해서 선택된 Peer 리스트를 제외하고, “and” 연산자의 Rlist인 hp를 가지고 있는 Peer 리스트와 조합한다. 그리고 Q2의 질의와 같이 Filter안에 존재하므로 하나의 Sub 쿼리가 묶여서 namecard를 질의한 Peer 리스트와 조합한다.

4. 결론 및 향후 연구 방향

본 연구에서는 JXTA를 기반으로 XML 문서를 효율적으로 검색할 수 있는 시스템을 설계하고 구현하였다. JAVA로 구현 하였으므로 모든 OS에 독립적으로 동작하였고, 모든 Peer에 질의하지 않고 선택적으로 질의함으로써 응답시간을 줄이고 질의에 대한 응답의 정확성을 높일 수 있었다. 최근 P2P 환경의 다양한 애플리케이션 서비스가 증가하면서 P2P 기반의 여러 검색 엔진들이 발표되고 있다. 하지만, 효율적인 검색이 이루어지지 못하고 있는데 본 논문은 이러한 P2P 환경 안에서 후보 Peer를 선정하고, XML 구조 정보를 이용하여 효과적인 검색이 가능하도록 노력하였다. 하지만, XPath 질의를 수행하는데 있어 질의의 횟수가 키워드의 개수만큼 증가하고, 복잡한 XPath 질의 처리에는 효율적이지 못했다.

본 논문은 XPath 질의를 하기 전에 키워드 질의를 조합함으로써 수행하는 키워드 검색 횟수를 감소시키고, XPath에서 제공하는 다양한 함수와 연산자들의 처리 등의 향후 연구가 필요하다.

참고문헌

- [1] Extensible Markup Language(XML) 1.0, "<http://www.w3.org/TR/1998/REC-xml>", 1998
- [2] J. Robie, J.Lapp, D.Schach, XML Query Language(XQL) Proposal, "<http://www.w3.org/Style/XSL/Group/1998/09/XQL-proposal.html/REC-xml>", 1998.
- [3] Daniel Brookshier, Darren Govoni, Navaneeth Krishnan, "JXTA: Java P2P Programming", SAMS, 2001.
- [4] 최지원, "IT 핵심용어- P2P", "<http://www.zdnet.co.kr>", 2001.
- [5] C.T. chen, F.Wang, G.Maini, L.Fransco-Revilla, "MUSE : Music Search Engine", CPSC-670, 2000.
- [6] 허명식, 손기락, "XQL을 지원하는 XML 문서 저장 시스템의 설계 및 구현", 정보과학회 학술발표 논문집(I),1999.
- [7] Sing Li, "JXTA Peer-to-peer Computing with Java", Wrox,2001.
- [8] L. Galanis, Y. Wang, S.R. Jeffery, D.J. Dewitt, "Processing XML Containment Queries in a Large Peer-to-Peer System", Proceedings of the 28th VLDB Conference, 2002.