XML 문서의 효율적인 경로 통합 기법

이범석, 황병연
가톨릭대학교 컴퓨터공학과

e-mail : {bslee, byhwang}@catholic.ac.kr

An Efficient Path Combining Strategy of XML Document

Bum-Suk Lee, Byung-Yeon Hwang
Dept. of Computer Engineering, The Catholic University of Korea

요 약

XML은 비즈니스 메시징, 웹사이트 정보 통합, 그리고 카탈로그 통합 등의 분야에서 다양한 데이터를 표현하기 위한 포맷으로 급격하게 성장했다. 그러나 XML 데이터의 형태가 고정되어 있지 않기 때문에 전통적인 질의 방법이 항상 정확한 결과를 보여주지는 않는다. 또한 객체 지향 DBMS가 이 영역에 적합한지의 여부는 아직 명확하지 않다. 따라서 XML 데이터를 효율적으로 검색하기 위해 기존의 관계형 DBMS와 연계하여 구조 유사성을 기반으로 하는 검색 기법이 연구되고 있다. 그 중 문서, 경로, 단어로 구성된 3차원 비트맵 인덱스를 이용한 검색 시스템은 다른 XML 문서 검색 시스템보다 훨씬 빠른 수행 속도를 보여주지만, 3차원의 메모리 구조를 사용하여 많은 저장공간을 필요로 하는 단점이 있다.

본 논문에서는 XML 문서를 저장할 때 경로들 사이의 유사성을 이용하여 XML 데이터의 경로를 통합하는 기법에 대해 소개한다. 이렇게 통합된 경로를 이용하여 생성하는 3차원 비트맵 인덱스는 그 크기가 상당히 줄어들게 되고, 기존의 연구에서 보여주었던 문제점들을 해결하게 되었다.

1. 서론

인터넷의 발전에 가장 큰 영향을 주었던 것은 바로 웹이며, 이 웹상에 문서를 기술하는 방법은 HMTL(Hyper Text Markup Language)이었다. HTML은 누구나 사용할 수 있을 만큼 간단하고, 보편적이며, 특별한 데이터 타입이 사용되지 않고 단순한 텍스트이기 때문에, 이식성과 사용이 편리하다는 장점이 있다. HTML은 이러한 여러 가지 장점에도 불구하고, 정해진 DTD 때문에 지정되지 않은 태그의 사용이 불가능 하다는 점과 문서의 구조를 고려하지 않고, 문서의 표현양식으로 만들어졌다는 단점을 가지고 있다[1]. 날로 발전하는 인터넷 시스템에 있어서 사용자의 요구가 다양해짐에 따라 이러한 필요를 충족시킬 수 있는 다른 방안이 필요하게 되었다. 그래서 XML(Extensible Markup Language)[2]이라는 새로운 문서 표준이 나오게 되었다. XML은 HTML과는 달리 데이터의 내용과 구조를 모두 표현하는 자기 서술적 특징을 가지고 있어, 새로운 웹 기반 애플리케이션 개발 시 데이터의 표현 및 교환의 표준으로서 각광을 받고 있다. 이러한 변화에 따라 XML과 관련된 다양한 연구의 필요성이 증대되고 있는데, 특히 XML로 표현된 문서를 효과적으로 저장, 접근, 검색하기 위한 저장 관리 시스템에 대해 많은 연구들이 활발하게 진행되어왔다[3].

이전의 저장 관리 시스템은 문서의 내용 정보에만 중점을 두어 문서를 저장 관리하였기 때문에, 주로 내용 색인만을 사용하였으나, 최근의 XML과 같은 구조의 문서는 문서의 내용뿐만 아니라, 그러한 내용이 무엇과 관련되어 있는지에 대한 구조 정보도 하나의 문서에 같이 포함하고 있기 때문에, 이러한 구조 정보를 내용 정보와 같이 손실 없이 저장 처리해 줄 수 있는 저장 관리기가 필요하다. 최근까지 이러한 구조문서의 저장과 검색에 대한 많은 연구[4,5,6,7]가 점차 늘어나고 있다.

BitCube[8,9]는 XML 검색에서 빠른 검색 속도를 통해 뛰어난 성능을 입증한 3차원 비트맵 인덱스 기법이다. 본 논문에서는 BitCube가 클러스터 내의 문서의 증가에 따라 급격하게 인덱스의 크기가 증가하게 되고, 이로 인해 연산 수행속도가 저하되는 문제점을 해결하기 위해 XML 문서의 경로 통합 기법을 제안한다. 이 기법은 XML 문서를 DBMS에 저장할 때, 추출한 경로를 통합하여 저장하고, 기존 비트맵 인덱스에서 단어로 구성했던 한 축 대신 value를 한 축으로 저장한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 기술한다. 3장에서는 XML 문서의 경로 통합 기법에 대해 소개하고, 4장에서는 개선된 비트맵 인덱스의 성능평가를 제시한다. 마지막으로 5장에서는 결론 및 향후 연구 계획에 대해 기술한다.

2. 관련 연구

2.1 3차원 비트맵 인덱스와 그 구성

3차원 비트맵 인덱싱인 BitCube는 다른 인덱싱 기법과는 달리 인덱스를 트리 구조로 구성하지 않는다. 이것은 빠른 검색을 위해 기존의 인덱싱 기법이 트리의 노드를 단위로 탐색을 수행하는 것과 달리 XML 문서의 경로를 추출하고 이 경로를 단위로 탐색을 수행한다. 또한 BitCube는 XML 문서에서 효과적으로 정보를 추출하기 위해 Bit-wise 연산이 가능한 3차원 비트맵 인덱스를 사용한다.

하나의 인덱스는 특정 클러스터를 의미하고 이 클러스터는 유사한 구조의 문서들이 모여 있는 집합이라 할 수 있다. 각각의 클러스터는 자신에게 포함된 문서와 그 문서에 포함된 경로, 그리고 그 경로에 포함된 단어를 사용하여 3차원 비트맵 인덱스를 구성한다. BitCube는 유사한 구조의 문서들을 적절한 클러스터에 수집하기 위해 경로를 중심으로 클러스터링을 수행한다. 예를 들어 시스템에 어떤 문서가 삽입된다면 시스템은 삽입된 문서와 기존의 클러스터들 사이의 구조적인 유사도를 측정하고 가장 유사도가 높은 클러스터에 그 문서를 삽입하고 인덱스를 새롭게 갱신한다. 여기서 유사도는 특정 문서가 갖고 있는 경로들이 얼마나 고르게 클러스터에 분포되어 있는지를 나타내는 정도이다. 각각의 클러스터는 유사한 경로를 많이 포함하는 문서들의 집합이므로 클러스터 내의 문서들이 갖고 있는 보편적인 경로들은 해당 클러스터를 대표하는 특징이라 볼 수 있다. 이 특징들의 집합을 클러스터의 중심이라 한다. 그러므로 삽입된 문서는 그것이 내포하고 있는 경로들과 클러스터의 중심과의 구성을 비교하고 가장 구성이 유사한 클러스터에 삽입된다.

BitCube는 기존의 XQEngine, XYZFind와 같은 시스템들과의 성능 평가에서 빠른 검색 속도를 통해 이미 뛰어난 성능을 입증하였다[8,9].

2.2 BitCube의 문제점

BitCube는 빠른 검색 성능으로 뛰어난 성능을 입증하였으나 문서의 증가에 따라 인덱스의 크기가 급격하게 늘어나는 단점을 지니고 있다. 이는 클러스터 내의 문서의 개수 증가에 따라 메모리 사용량이 굉장히 높아지는 것을 의미하며, 연산 수행 속도를 저하시키는 원인이 된다.

이런 문제의 원인은 BitCube의 인덱스 크기 증가가 평면에 의존적이기 때문이다. 어떤 문서나 경로가 삽입될 때, 이것들이 이미 클러스터 내에 존재한다면 인덱스를 확장하지 않지만, 그렇지 않은 경우에는 인덱스의 크기가 기존의 인덱스의 한 평면만큼 확장된다. 다시 말해 추가된 문서에 존재하는 어떤 한 경로가 기존에 구성된 비트맵 인덱스에 존재하지 않는다면, BitCube는 기존 인덱스에 존재하는 모든 문서의 개수와 단어의 개수를 곱한 만큼의 공간을 필요로 하게 된다. 그러므로 BitCube의 인덱스 크기 증가는 클러스터 내의 문서의 개수가 증가할수록 급격하게 늘어나게 된다. 게다가 클러스터 단위로 인덱스를 메모리에 적재하는 시스템인 BitCube는 단일 클러스터의 인덱스 크기가 지나치게 커지는 경우 연산의 성능 저하는 급격하게 증가할 수 있으며, 극단적인 경우 시스템이 연산을 전혀 수행할 수 없는 상황이 발생할 수도 있다.

3. XML 문서의 경로 통합 기법

본 장에서는 BitCube의 문제점을 해결하기 위해 XML 문서의 경로를 통합하는 기법에 관하여 기술한다. 제안하는 기법은 유사한 경로를 통합하고, BitCube의 단어 축을 value 축으로 구성하여 기존의 문제점을 개선한다. 이 방법으로 통합된 경로의 정보를 사용하여 BitCube를 생성하면, 기존의 방법보다 메모리 사용량 면에서 훨씬 뛰어난 성능을 갖게 된다.

3.1 제안하는 시스템의 구조

제안하는 시스템을 효율적으로 구현하기 위해서

는 시스템 구조의 설계가 선행되어야 한다. 그림 1은 전체 시스템의 구조를 나타낸다.

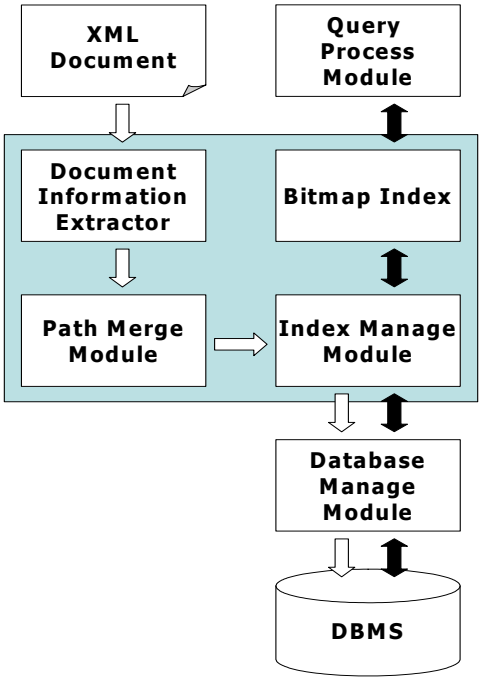

그림 1. 시스템 구조

이 시스템은 크게 두 개의 과정으로 나누어지는데, 첫 번째 과정은 문서 입력 과정이다. 새로운 XML 문서가 입력되면 해당 문서의 정보를 추출하여, 유사한 경로를 통합한다. 이 정보들은 인덱스 관리 모듈에서 BitCube를 구성하고, DBMS에 저장이 된다. 두 번째 과정은 질의 처리 모듈이다. 질의가 입력되면 BitCube에서 검색을 하여 DBMS에 있는 정보를 반환하게 되는데, 이때 DBMS에 저장된 정보를 이용하여 결과를 사용자에게 보여주는 과정을 질의 처리 모듈이 담당한다.

3.2 유사 경로 통합

XML 문서 경로를 통합하여 작고 빠른 인덱스를 구성하는 방법은 A(k)-Index[10], 1-Index[11], 그리고 DataGuide[12]와 같은 것들이 있는데, 이들은 그래프 기반 인덱스 기법으로 3차원 비트맵 인덱스에 적용시키기는 힘들다.

본 논문에서 제안하는 방법은 모든 조상 노드의 경로가 같은 형제 노드를 병합한다. 이렇게 병합된 노드를 이용해 1과 0으로 표현되는 비트맵 인덱스를 구현할 때에 병합된 두 개의 경로 중 하나의 경로만 존재해도 1로 표현한다. 또한 본 논문에서 XML 문서의 속성(attribute)은 해당 노드의 자식 element로 표현하고 저장할 때, 노드 이름에 @를 표시한다. 그림 2는 XML 문서의 예제이다.

```
[example.xml]

<e0>
   <e1>v1</e1>
   <e2>
      <e3>v2 v3 v4 v5 v19 v20</e3>
      <e4 a1="v7">
         <e6>v8 v9 v10 v2 v6 v11</e6>
         <e7>v12 v13 v17 v18</e7>
      </e4>
   </e2>
   <e8>v14 v15 v16 v4 v7 v8</e8>
</e0>
```

그림 2. 예제 XML 문서

example.xml에서 e0.e1과 e0.e8은 서로 형제 노드이다. 또한 e4의 속성인 a1은 e0.e2.e4.@a1로 표현하여 저장하면, e0.e2.e4.e6과 e0.e2.e4.e7과 함께 형제 노드가 된다. 이들은 각각 e0.(e1|e8)로, e0.e2.e4.(e6|e7|@a1)로 저장된다.

example.xml을 기존의 방법을 이용하여 비트맵 인덱스를 구현하려면, d × p × w 만큼의 메모리를 사용하므로, 1개의 문서, 6개의 경로, 20개의 단어가 추출되어, 총 120bit의 메모리를 사용하게 된다. 그러나 제안하는 기법을 이용하면, 3개의 경로, 7개의 value를 사용하므로, 21bit의 메모리를 사용한다. 이와 같은 단순한 예에서도 알 수 있듯이 사용하는 메모리의 크기가 크게 줄어드는 것을 보여준다.

4. 성능 평가

본 연구의 성능 평가는 기존의 방법과 제안하는 방법의 3차원 비트맵 인덱스를 구성할 때, 사용하는 메모리의 크기 비교에 대해 수행하였다. 성능 평가에 사용된 시스템의 사양은 Windows 2000 Server의 PC에 MS-SQL 2000, 그리고 프로그램의 구현은 JAVA 버전 1.4.2.07과 JDOM 버전 1.0[13]을 이용하였다.

실험을 수행하기 위해, 임의의 XML 문서 100개[14]를 Document Information Extractor에서 처리하였다. 기존의 방법으로 추출된 경로의 개수는 7,842개, 단어의 개수는 25,320개였다. 이때 사용되는 메모리의 크기는 d × p × w / 8 (byte)의 식에 따라, 2,481,993,000byte 이다. 제안하는 방법을 적용하여 문서를 추출하였을 때, 경로의 개수는 1,018개, value의 개수는 7,842개이다. 이러한 정보를 가지고

3차원 비트맵 인덱스를 구성한다면, 사용되는 메모리는 99,789,450byte 이다. 이러한 결과는 제안하는 방법이 문서의 구조에 따라 정도의 차이는 있겠지만, 평균적인 메모리 사용량을 4% 정도로 줄인 것이다. 다시 말하면 제안하는 방법은 기존의 방법에 비해 약 96%의 메모리 효율을 갖게 된다.

이 밖에도 위의 데이터를 이용하여 수행 속도에 대한 실험을 수행하였으나, 100개의 문서로는 두 방법 모두 큰 차이가 나지 않았다. 이것은 질의의 내용에 따라 다른 성능을 보였고, 평균적으로는 테이블 검색이 줄어들게 된 새로운 기법이 20ms 정도 더 빠른 성능을 보였다. 이러한 차이는 Path 테이블의 행 개수의 차이, 단어와 value 테이블의 행 개수 차이에서 기인한 것으로 보인다.

5. 결론 및 향후 연구 계획

기존의 BitCube는 문서가 삽입될 때, 한 개의 경로가 늘어난다면, d × w 만큼의 bit가 확장되어, 새로운 경로가 많아지면, 사용되는 메모리도 평면 의존적으로 급격하게 늘어나는 문제점을 갖고 있었다. 또한 이 인덱싱 기법은 연산을 수행하기 위해서 속도가 매우 빠른 Bit-wise연산을 사용하지만, 실제 결과를 추출하고 반환하기 위해서는 3차원 비트맵 인덱스의 한 평면 전체에 대하여 이 연산을 수행해야만 한다. 그러므로 인덱스의 크기가 연산의 수행 속도에 직접적인 영향을 미치게 된다.

본 논문에서 제안된 경로 통합 및 value를 사용하는 기법은 인덱스의 크기를 줄이는데 매우 효율적인 결과를 보여주었다. 이처럼 메모리 사용량을 줄이는 결과는 BitCube가 클러스터 단위로 인덱스를 메모리에 적재하는 시스템임을 고려할 때, 단일 클러스터의 인덱스 크기가 지나치게 커지는 경우 메모리의 적재 한계에 의해 시스템이 연산을 전혀 수행할 수 없는 문제에 대해서도 부분적인 해결책을 제시할 수 있다.

향후 연구에서는 질의된 경로의 유사 경로를 함께 사용자에게 보여줄 수 있는 BitCube에 대해 연구를 진행하고자 한다. 또한 문서의 개수가 아무리 많아져도 시스템이 멈추지 않게 하는 방법에 대한 연구 수행을 계획하고 있다.

참고문헌

[1] 연제원, 김상균, 이규철, 나중찬, 김명준, “XML 문서의 효율적 검색 및 변경을 위한 저장관리기의 설계 및 구현,” 한국전자통신연구원, 2000.

[2] W3C, “Extensible Markup Language(XML) Version 1.0 (Second Edition),” http://www.w3c.org/TR/REC-xml, October 2000.

[3] S. Ceri, P. Fratenali and S. Paraboschi, "XML: Current Developments and Future Challenges for the Database Community,” Proc. of the 7th Int'l. Conf. on EDBT, pp. 3-17, March 2000.

[4] R. Goldman, J. McHugh, and J. Widom, "From Semistructured Data to XML: Migrating the Lore Data Model and Query Language,” Proc. of 2nd Int'l. Workshop on the Web and Database, 1999.

[5] C. C. Kanne and G. Moerkotte, "Efficient Storage of XML Data," Technical Report, University of Mannheim, 1999.

[6] S. Kim, Y. Kim, J. Lee, and H. Lim, "Developing a Native Storage Structure for XML Repository System in Main Memory," Proc. of the 5th IEEE Int'l. Conf. on High Speed Networks and Multimedia Communications, pp. 96-100, July 2002.

[7] C. Chan and Y. Ioannidis, "Bitmap Index Design and Evaluation," Proc. of ACM SIGMOD Conf., Seattle, pp.355-366, June 1998.

[8] J. P. Yoon, V. Raghavan, V. Chakilam, and L. Kerschberg, “BitCube: A Three-Dimensional Bitmap Indexing for XML Documents,” Journal of Intelligent Information System, Vol.17, pp.241-254, 2001.

[9] J. Yoon, V. Raghavan, and V. Chakilam, "BitCube : Clustering and Statistical Analysis for XML Documents," 13th Int'l. Conf. on Scientific and Statistical Database Management, Virginia, July 2001.

[10] R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes, "Exploiting Local Similarity for Indexing Paths in Graph-Structured Data," 18th IEEE Int'l. Conf. on Data Engineering, pp.129-140, 2002.

[11] T. Milo and D. Suciu, "Index Structures for Path Expressions," 7th Int'l. Conf. on Database Theory, 1999.

[12] R. Goldman and J. Widom. “Approximate DataGuides,” Proc. of the Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats, pp.436-445, 1999.

[13] http://www.jdom.org

[14] http://us.imdb.com/top_250_films