

# 객체 관계 데이터베이스 시스템과 하이브리드 오더 인코딩을 이용한 XML 저장 시스템 설계 및 구현\*

김영우<sup>o</sup> 홍의경  
서울시립대학교 컴퓨터과학부  
{ywkim<sup>o</sup>, ekhong}@venus.uos.ac.kr

## Design and Implementation of XML Storage System using Object Relational Database System and Hybrid Order Encoding Method

Youngwoo Kim<sup>o</sup> Eui Kyeong Hong  
Department of Computer Science, University Of Seoul

### 요 약

인터넷의 발전은 다양한 데이터의 폭발적인 증가를 가져왔다. 유연하고 효과적인 데이터 표현 능력을 지닌 XML이 인터넷 환경에서 데이터 표현 및 교환 수단으로 여러 분야에서 표준으로 활용되고 있다. 그래서 대용량 XML 문서의 저장 및 관리에 대한 연구의 필요성이 증가하였다. 현재 다양한 XML 저장 기법과 XQuery를 이용한 XML 질의 처리에 대한 연구가 활발히 진행되고 있다.

본 논문에서는 객체 관계 데이터베이스 시스템을 이용하여 대용량 XML 문서 처리에 적합하도록 XML 저장 시스템을 설계 및 구현하였다. 또한 하이브리드 오더 인코딩 기법을 이용하여 저장된 XML 문서의 갱신 성능과 XML 질의 처리 성능을 개선하였다. 그리고 Xbench를 이용하여 생성한 대용량 XML 문서로 XML 저장 시스템의 성능을 평가하고 분석하였다.

### 1. 서 론

최근 IT분야의 컴퓨팅 환경은 과거 1980년대부터 1990년대 중반까지 널리 활용되던 클라이언트-서버(Client-Server) 환경에서 인터넷 환경의 웹(World Wide Web)을 기반으로 한 분산 컴퓨팅(Distributed Computing) 환경으로 변화하고 있다. 그에 따라 다양한 데이터의 양이 급속하게 증가되었고, 이러한 데이터를 효율적으로 표현하고 저장, 관리하기 위한 연구가 진행되었다. 그러한 연구의 중심에 XML(Extensible Markup Language)이 있다. XML[1]은 W3C(World Wide Web Consortium)에서 제안한 인터넷 환경에서 데이터 표현과 교환을 위한 표준 마크업 언어이다. XML은 기존에 많이 사용했던 마크업 언어인 HTML과 SGML의 장점을 수용하여 데이터 표현 능력이 우수하며 사용하기 쉽고 확장성이 뛰어나다. 그래서 현재 XML은 다양한 분야에서 표준으로 활용되고 있다. 또한 XML 문서로부터 정보를 검색하기 위한 질의 처리 연구로 다양한 질의어가 제안되었다. 현재는 XPath와 XQuery가 주로 사용되고 있으며, XPath가 XQuery로 흡수되어 XQuery가 XML 질의 언어의 표준으로 사용되고 있다.

XML 문서를 효율적으로 저장하고, 필요한 정보를 검색하고 관리하기 위하여 다양한 방법들이 제안되었다. 그중에서 XML 문서를 데이터베이스 시스템에 저장하고 관리하려는 연구가 활발히 진행되었다[2]. XML 문서를 데이터베이스 시스템에 저장하고 관리하면 기존에 연구된 데이터베이스 시스템의 기능이나 질의 처리 기능을 이용할 수 있는 장점이 있다[3].

본 논문에서는 객체 관계 데이터베이스 시스템을 이용하여 대용량 XML 문서의 삽입, 수정, 검색이 용이하도록 하이브리드 오더 인코딩을 이용한 XML 저장 시스템을 설계 및 구현하였다.

본 논문의 구성은 다음과 같다. 2절에서는 데이터베이스 시스템을 이용한 XML 저장 기법들에 대하여 살펴보고, 3절에서는 본 논문에서 제안한 XML 저장 시스템의 내부 저장 구조와 XML 질의 처리 설계, 실제 시스템 구현에 대하여 서술한다. 4절에서는 대표적인 XML 질의 유형에 대한 XQuery 성능과 갱신, 라운드 트리핑 성능을 평가하고 분석한다. 마지막으로 5절에서는 결론과 향후 연구 방향에 대하여 기술한다.

### 2. 관련 연구

#### 2.1 XML 저장 기법

XML 문서를 저장하고 관리하기 위하여 파일 시스템, (객체)관계 데이터베이스 시스템을 이용한 방식, 네이티브 XML 데이터베이스 등이 연구되었다. 파일 시스템을 이용한 기법은 확장성이나 질의 처리에 제한이 많고 네

\* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

이터브 XML 데이터베이스 시스템의 경우 많은 부분에서 기술적 성숙이 이루어지지 못하였다. 그래서 기존에 많이 연구되고 기술적으로 성숙한 (객체)관계 데이터베이스 시스템을 이용한 방식이 활발히 연구되었다. 기존 데이터베이스 시스템을 이용하여 XML 문서를 저장하고 관리하면 질의 처리, 트랜잭션, 권한 시스템, 인덱스 기능 등 기존 데이터베이스 시스템의 기능을 활용할 수 있다.

기존 데이터베이스 시스템을 이용한 XML 저장 기법들은 접근 방식과 데이터 모델에 따라 분류할 수 있다[3]. 일반적으로 XML 문서는 텍스트 중심 XML 문서와 데이터 중심 XML 문서로 나눌 수 있다. [2]에서는 이러한 XML 문서의 특징에 따라 가능한 저장 기법을 구분하였다. XML 문서 전체를 하나의 BLOB(Binary Large Object)이나 CLOB(Character Large Object) 형태로 저장하거나 XML 문서의 노드를 데이터베이스의 릴레이션과 애트리뷰트로 나누어 저장하는 분할 저장 기법이 주로 사용된다. 분할 저장 기법은 XML 문서가 데이터베이스에 저장되는 구조에 따라 XML 문서의 XML 스키마나 DTD를 이용하는 스키마 종속적인 기법과 스키마 독립적인 기법으로 구분할 수 있다. 스키마 종속적인 기법은 XML 스키마에 따라 데이터베이스에 생성된 데이터베이스 스키마가 비효율적으로 모델링될 수 있는 단점이 있다. 그리고 XML 스키마나 DTD가 없는 문서가 스키마가 존재하는 문서보다 많으므로 본 논문에서는 XML 문서의 스키마와 관계없이 데이터베이스의 스키마를 정의하여 XML 문서의 정보를 저장하는 스키마 독립적인 저장 기법을 사용한다. 또한 다양한 XML 문서를 처리하기 위하여 XML 문서의 노드마다 오더 레이블링을 사용하여 텍스트 중심 XML 문서와 데이터 중심 XML 문서를 동시에 처리할 수 있도록 하였다.

XML 문서의 노드 오더 인코딩 기법으로는 글로벌 오더, 로컬 오더, Dewey 오더가 있다[4]. 글로벌 오더의 경우 XML 문서의 노드 순서대로 레이블링하는 방식으로 질의 처리 성능이 우수한 장점이 있지만 노드의 삽입, 삭제가 발생할 경우 노드의 레이블링을 다시 수행하여야 하는 단점이 있다. 로컬 오더는 노드의 갱신 성능이 우수하지만 노드들 간의 부모-자식 관계를 파악하기 어려운 단점이 있다. Dewey 오더와 같은 하이브리드 오더 기법은 갱신 연산에 따른 오버헤드를 줄이고 질의 처리 성능을 유지시키는 장점을 가진다[5].

### 3. XML 저장 시스템 설계 및 구현

#### 3.1 XML 저장 시스템 구조

전체 시스템은 그림 1과 같이 하부 저장 시스템으로 객체 관계 데이터베이스 시스템을 사용한다. XML 문서의 입력, 수정, 삭제, 질의 처리를 담당하는 XML 미들웨어는 인터페이스 모듈, 저장 관리 모듈, 질의 변환 모듈, XML 질의 처리 모듈로 구성된다. XML 문서의 정보를 객체 관계 데이터베이스 시스템에 저장하기 위하여 그림 2와 같이 스키마를 작성하였다. 스키마는 노드의 경로 정보와 실제 데이터를 같은 릴레이션에 저장하는 인라이닝(Inlining) 기법[6]을 사용하였다. 저장 단계에서는 XML 문서의 각 노드가 그림 3과 같이 인코딩된 후 각 자리에 '0'을 맞추어 저장한다. 예제 XML 문서

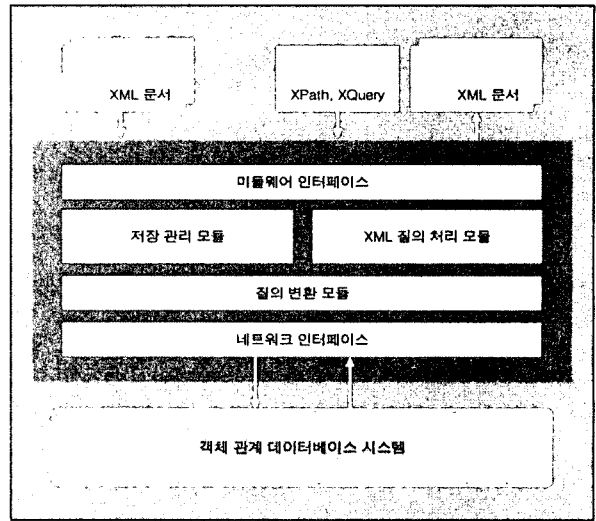


그림 1. XML 저장 시스템 구조

order.xml을 트리 형태로 모델링한 후 오더 레이블링 간격을 '1'로 설정하여 모델링하였다. 이 간격은 문서의 성질에 따라 달라질 수 있다. 질의 처리 단계에서는 수행할 XPath나 XQuery를 SQL로 변환한 후에 객체 관계 데이터베이스 시스템으로 질의를 수행한다. XQuery 질의를 처리하기 위해서 경로 탐색이 필요하다. 경로 탐색은 'NodeOrder'와 'NodeType'으로 식별이 가능하다. 그림 3에서 노드 오더가 '1.3'인 'lineitem'의 자식 노드 중 엘리먼트를 검색할 경우 'NodeOrder'가 '1.00003.x'인 노드 중 'NodeType'이 'ELEMENT'인 노드만 검색하면 된다. 애트리뷰트의 경우도 엘리먼트와 동일하다.

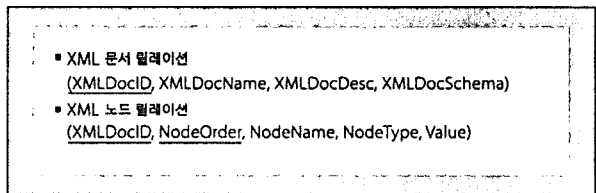


그림 2. XML 저장 구조 스키마

#### 3.2 XML 저장 시스템 구현

본 논문의 XML 저장 시스템은 Oracle Database 10g 객체 관계 데이터베이스 시스템을 사용하였다. XML 응용 프로그램과 미들웨어는 JAVA로 작성되었으며 SUN Microsystems의 J2SE 1.4.2 버전을 사용하였다. XML 파서로는 Apache Software Foundation의 Xerces Java 2를 사용하였다. 노드 인코딩 과정에서 XML 파싱은 DOM 방식보다 대용량 XML 문서 처리에 적합한 SAX 방식을 사용하였다.

### 4. XML 저장 시스템 성능 평가

XML 저장 시스템의 성능 평가는 [7],[8]에서 제안한

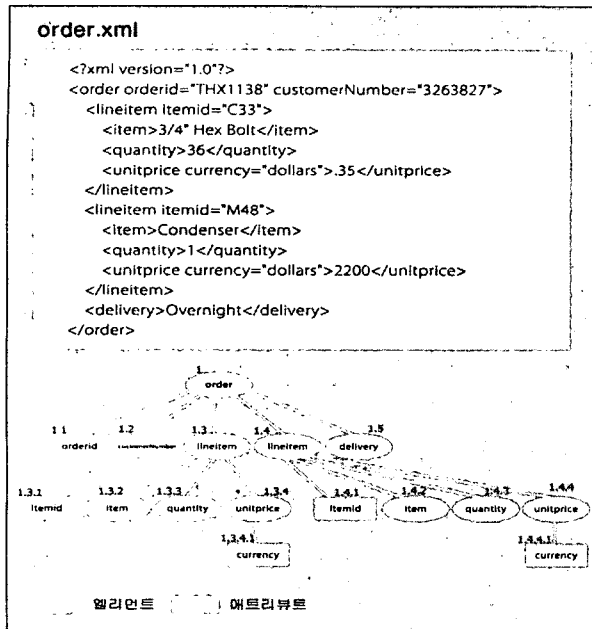


그림 3. XML 노드 오더 인코딩

글로벌 오더 인코딩과 분할 테이블 기법을 이용한 XML 저장 시스템과 성능을 비교, 평가하였다. 성능 평가에는 Xbench[9]를 사용하여 생성한 10 MB의 XML 문서를 사용하였고 펜티엄 4, 1.8 Ghz, 512 MB 메인 메모리로 구성된 시스템에서 수행하였다.

성능 평가는 XML 라운드 트리핑 성능, XML 질의 성능, 노드의 갱신 성능에 대하여 수행하였다. 실험 1은 XML 문서를 저장하고 전체 문서를 다시 불러오는 라운드 트리핑 성능을 평가하였다. 실험 2는 XQuery에 나타나는 경로 탐색 질의 성능 평가를 수행하였다. 실험 3에서는 저장된 XML 문서의 노드 삽입, 삭제를 통한 갱신 성능을 비교하였다. 성능 평가 결과는 표 1, 2, 3과 같다.

표 1. XML 문서의 라운드 트리핑 성능 비교 (실험 1)

글로벌 오더 + 분할 테이블	하이브리드 오더 + 인라이닝
642 sec	97 sec

표 2. 경로 탐색 질의 성능 비교 (실험 2)

글로벌 오더 + 분할 테이블	하이브리드 오더 + 인라이닝
41 sec	8 sec

표 3. XML 문서의 노드 갱신 성능 비교 (실험 3)

글로벌 오더 + 분할 테이블	하이브리드 오더 + 인라이닝
210 sec	6 sec

성능 평가 결과로부터 본 시스템이 기존 시스템과 비교하여 인라이닝 기법을 통해 과도한 조인연산을 피함으로써 질의 성능 향상을 보였다. 또한 하이브리드 오더를 사용함으로써 노드의 범위를 벗어나지 않는 경우에는 갱신 연산에 따른 추가 레이블링이 없기 때문에 XML 문서의 갱신에도 좋은 성능을 보였다.

### 5. 결론 및 향후 연구 방향

본 논문에서는 객체 관계 데이터베이스 시스템을 기반으로 하이브리드 오더 인코딩을 이용하여 대용량 XML 문서의 저장과 갱신, 질의 처리에 적합한 XML 저장 시스템을 설계하고 구현하였다.

향후 연구로는 다양한 XQuery의 기능을 지원할 수 있도록 질의 처리 모듈의 개선과 질의 변환 단계에서 질의 최적화에 대한 연구가 필요하다.

### 참고문헌

- [1] T. Bray, et al., "Extensible Markup Language (XML) 1.0," W3C Recommendation 04, Feb. 2004.
- [2] I. Mlynkova and J. Pokorny, "XML in the World of (Object-)Relational Database Systems," Proc. of the XIII. Int'l Conf. ISD, Vilnius, Lithuania, Sept. 2004.
- [3] H. Vieira, et al., "XVerter: Querying XML data with OR-DBMS," Proc. of the 5th ACM Int'l Workshop on Web Information and Data Management, New Orleans, Louisiana, pp.37-44, 2003.
- [4] I. Tatarinov, et al., "Storing and Querying Ordered XML using a Relational Database System," Proc. of ACM SIGMOD, pp.204-215, 2002.
- [5] P. O'Neil and E. O'Neil, "ORDPATHs: Insert-Friendly XML Node Labels," Proc. of ACM SIGMOD, Paris, France, pp.903-908, 2004.
- [6] D. Florescu and D. Kossmann, "Storing and Querying XML Data using an RDBMS," IEEE Data Engineering Bulletin 22(3), pp.27-34, 1999.
- [7] 고영기, 홍의경, "분할 저장 시스템에 적합한 XPath 질의 처리기의 설계," 한국정보과학회 가을학술대회 논문집, Vol. 29, No. 2, pp.52-54, 2002.
- [8] 장형화, 홍의경, "관계 데이터베이스 시스템 기반의 XQuery 질의 처리기 설계," 한국정보과학회 가을학술발표 논문집(II), Vol. 30, No. 2, pp.106-108, 2003.
- [9] B. B. Yao, M. T. Ozsu, and N. Khandelwal, "XBench Benchmark and Performance Testing of XML DBMSs," Proc. of 20th Int'l Conf. on Data Engineering, Boston, MA, pp.621-632, March 2004.