

XML 실체뷰를 이용한 XQL 질의 분할의 성능 분석*

문찬호⁰ 강현철
중앙대학교 컴퓨터공학과
moonch@dblab.cse.cau.ac.kr hckang@cau.ac.kr

Performance Analysis of XQL Query Decomposition Using XML Materialized Views

ChanHo Moon⁰ Hyunchul Kang
Dept. of Computer Science and Engineering,
Chung-Ang University

요약

XML 저장소 내에 XML 문서들과 그들로부터 도출된 XML 실체뷰가 있다고 가정할 때, XML 문서 검색의 성능 향상을 위해서 이들 실체뷰를 이용하여 질의를 처리할 수 있다. 즉, 하부 XML 문서에 대한 원래의 질의를 관련된 실체뷰에 대한 질의로 변환하여 수행함으로써 질의 응답시간을 줄일 수 있다. 실체뷰를 이용한 질의 처리의 유형으로는 (1) 실체뷰로부터 원하는 결과를 모두 얻을 수 있는 유형과 (2) 질의 결과의 일부는 실체뷰에 존재하지만 일부는 하부 XML 문서로부터 검색해야 하는 유형이 있다. 본 논문에서는 두번째 유형에 대하여 연구하였다. 주어진 질의를 (1) 실체뷰에 대한 질의와 하부 데이터에 대한 질의로 분할하여 처리한 후 두 결과를 통합하는 방법과 (2) 원래의 질의를 실체뷰를 이용하지 않고 처리하는 방법 간의 성능을 비교, 분석하였다.

1. 서론

뷰는 다양한 데이터 저장소(repository)에 저장된 이질적인 데이터의 통합과 데이터 여과 기능을 제공한다. 웹 데이터에서 많은 비중을 차지하고 있는 XML 데이터에 대해서도 뷰는 유용한 개념이다[1]. 뷰는 질의 처리 시 성능 향상을 위해 실체뷰(materialized view)로 유지할 수 있는데, 뷰가 실체뷰로 제공되면 관련된 질의의 효율적인 처리에 이용될 수 있다. 최근 방대한 양의 XML 웨어하우스에 대한 질의 결과를 XML 실체뷰로 유지하기 위한 기법들이 연구되고 있다[2].

그림 1은 다수의 XML 문서(하부 데이터)를 저장하고 있는 XML 저장소를 대상으로 질의 q의 처리 과정을 나타낸 것이다. (a)는 실체뷰를 제공하지 않는 기존의 XML 저장소를 대상으로 한 질의 처리를 나타낸 것이다. 이와는 달리 (b)는 XML 저장소 내 실체뷰를 이용한 질의 처리를 나타낸 것이다. 그림 1 (b)에서 보는 것처럼, 하부 데이터로부터 뷰 정의를 통해 도출된 XML 실체뷰들이 XML 저장소 내에 있다고 하자. 이와 같은 XML 저장소에 대해 질의 q가 주어

졌을 때, XML 저장소에 대한 질의 결과의 일부를 관련된 실체뷰로부터 얻을 수 있다면, 질의 변환을 통해 q를 실체뷰에 대한 질의 q'과 하부 데이터로부터 나머지 결과를 얻는 질의 q''으로 변환하여 수행함으로써 방대한 양의 XML 저장소 내 문서들에 대한 검색을 피할 수 있어 질의 응답 시간을 줄일 수 있다[3].

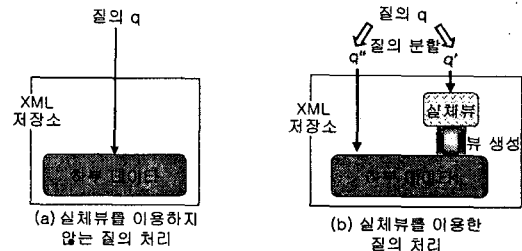


그림 1 XML 저장소를 대상으로 한 질의 처리

본 논문에서는 XML 저장소 내에 XML 실체뷰가 있다고 가정할 때, XML 문서에 대한 질의 결과의 내용 중 일부가 실체뷰에 존재할 경우, 주어진 질의를 (1) 실체뷰에 대한 질의와 (2) 실체뷰에 포함되지 않은 내용을 직접 하부 데이터로부터 검색하는 질의로 분할(decomposition)한 후 이들을 처리하여 그 결과를 통합하는 방법(이하, 실체뷰를 이용

* 본 연구는 한국과학재단 목적기초연구(R01-2000-00272)지원으로 수행되었음.

한 질의 처리)과 실체뷰를 이용하지 않고 처리하는 방법의 성능을 비교, 분석한다. 본 논문에서는 XML 질의어로 경로 표현식을 이용하여 질의 대상을 표현하는 XQL(XML Query Language)[4]을 사용하였다.

본 논문의 구성은 다음과 같다. 2절에서는 XML 실체뷰를 지원하는 XML 저장소의 구조 및 질의 분할과 질의 처리에 대해서 설명한다. 3절에서는 질의 분할에 대한 성능 분석 결과를 기술한다. 마지막으로 4절에서 결론을 맺는다.

2. XML 저장소 구조 및 질의 분할과 질의 처리

2.1 XML 저장소 구조

본 논문의 XML 저장소 구조는 객체 관계형 데이터베이스를 기반으로 하여 그림 2와 같이 크게 하부 데이터 영역과 XML 실체뷰 영역으로 나누어진다. 하부 데이터 영역은 XML 문서들을 엘리먼트 단위로 나누어 저장한 엘리먼트 테이블, 인덱스와 DTD 테이블로 구성된다. 엘리먼트 테이블은 DTD 식별자, XML 문서 식별자, 엘리먼트 식별자, 엘리먼트 이름, 경로 정보, 부모 엘리먼트 식별자, 자식 엘리먼트 식별자의 집합, 이전 형제 엘리먼트의 식별자, 다음 형제 엘리먼트의 식별자, 등의 필드로 구성된다. 인덱스는 엘리먼트 테이블을 구성하는 필드 중 엘리먼트 이름을 키로 갖는다. DTD 테이블은 DTD 정보와 DTD를 참조하는 XML 문서(들)의 루트('/') 엘리먼트가 엘리먼트 테이블 내에 어디에 위치하는지에 대한 정보가 저장된다. XML 실체뷰는 XML 문서 형태를 갖는다. 본 논문에서는 [3]과 [5]에서처럼 XML 실체뷰의 표현을 각 XML 하부문서로부터 XQL 질의 결과로 검색된 서브 트리들을 새로운 루트 엘리먼트로 묶은 것으로 정의하였다. XML 실체뷰 영역 또한 XML 실체뷰 정의로부터 얻어진 XML 실체뷰들을 엘리먼트 단위로 나누어 저장한 뷰 엘리먼트 테이블, 뷰 정의를 저장하는 뷰 정의 테이블과 DTD 테이블로 구성된다. 뷰 엘리먼트 테이블은 하부 데이터 영역의 엘리먼트 테이블과 같은 구조로 구성된다. 이때, 하부 데이터 영역의 인덱스는 엘리먼트 테이블 중 일부 엘리먼트의 인덱싱 정보만을 저장한다. 이는 하부 데이터 전체에 대한 인덱싱 정보를 유지하기 위한 저장 공간과 유지 비용이 크기 때문이다.

2.2 XQL 질의 모델

본 논문의 XML 뷰 내용은 XQL 질의 결과에 해당된다. XQL은 파일의 디렉토리 경로를 나타내는 것과 같은 경로 표현식을 이용하여 질의 대상이 되는 XML 문서의 목적 엘리먼트(target element)를 표시하고 조건을 표시하는 필터 연산자, 비교 연산자 등을 이용하여 질의 표현한다[4] XQL 질의의 결과 엘리먼트(result element)는 추출된 목적 엘리먼트와 그의 서브 트리(즉, 서브 엘리먼트들 모두)로 구성된다. 본 논문에서 고려한 XQL 질의는 다음을 만족한다고 가정하였다.

- 조건을 명시하기 위한 필터 연산은 한번만 사용한다.
- 필터 연산자 내의 조건이 사용하는 비교 연산자로는 '=' 과 'Sor\$'로 제한한다.
- 필터 연산 내에서 Sor\$ 연산 con1 Sor\$ con2가 주어졌을 경우(con1과 con2는 모두 '=' 연산을 이용한 조건), con1의 결과와 con2의 결과는 겹치지 않는다.

- 목적 엘리먼트는 하나이다. 단, 그 결과의 서브 엘리먼트는 모두 함께 추출된다.

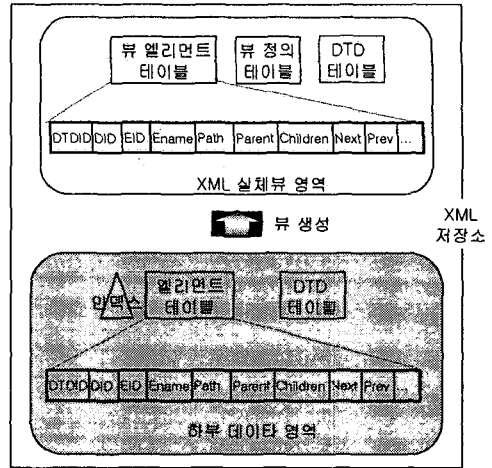


그림 2 XML 저장소의 구조

2.3 질의 분할

[3]과 [5]는 질의 분할 방법을 제시하고 있다. [3]에서는 XQL 질의 q와 실체뷰 v의 정의가 주어졌을 때, 질의 결과와 실체뷰 간의 포함관계에 따라 실체뷰를 이용한 질의 처리 유형을 (1) 실체뷰로부터 원하는 결과를 모두 얻을 수 있는 유형, (2) 질의 결과의 일부는 실체뷰에 존재하지만 일부는 실체뷰에 존재하지 않는 유형, 그리고 (3) 주어진 질의의 결과를 모두 하부 데이터로부터 얻는 유형으로 분류하고, 유형 (1)에 대하여 질의 q를 실체뷰 v에 대한 질의로 변환하는 알고리즘을 제시하였다. 그리고, [5]에서는 유형 (2)에 대하여 질의 q를 실체뷰 v에 대한 질의 q'과 하부 데이터에 대한 질의 q''로 XQL 질의를 분할하는 알고리즘을 모두 다섯 가지 경우로 나누어 제시하였다. 본 논문에서는 [5]의 질의 분할 알고리즘을 사용한다.

2.4 분할된 질의 처리

질의 q를 2.3절에서 언급한 질의 q'과 q''으로 분할하였을 때, 이들을 처리하는 과정(즉, XML 저장소 내의 엘리먼트 접근 과정)은 다음과 같다. 먼저, q'의 처리를 위해서는 검색 조건에 해당하는 엘리먼트의 엘리먼트 테이블 내 위치를 찾은 후(이때, 조건 엘리먼트에 대한 인덱스가 있을 때에는 인덱싱을 통해 위치를 찾을 수 있지만, 없을 때에는 하부 XML 문서 전체를 검색하여 조건 엘리먼트의 위치를 찾아야 함), DTD 테이블을 참조하여 이 엘리먼트를 포함하고 있는 XML 문서의 루트 엘리먼트 위치를 엘리먼트 테이블에서 찾는다. 그리고, 검색된 루트 엘리먼트로부터 연결 정보(즉, 2.1절에서 설명한 경로 정보, 부모 엘리먼트 식별자, 자식 엘리먼트 식별자의 집합, 이전 형제 엘리먼트의 식별자, 다음 형제 엘리먼트의 식별자 등)를 이용하여 목적 엘리먼트와 그의 서브 트리를 추출한다. 한편, q''의 처리를 위해서는 실체뷰를 접근해야 하는데, 본 논문에서는 실체뷰의 내용 전체가 q'의 결과가 되는 질의 q의 처리만을 고려하였다. 실체뷰의 내용 전체 또는 일부를 주어진 질의 결과의 일부로 사용(즉, q'의 결과로 사용)하기 위해서는 실체뷰의

일관성이 유지되어야 한다. 본 논문에서는 [2]에서 제시된 하부 XML 문서의 변경을 XML 실체부에 점진적으로 지연(deferred) 반영하는 기법을 사용한다.

한편, q' 의 결과가 실체부의 내용 전체이므로, q' 과 q'' 의 결과 통합은 q' 의 결과인 실체부의 루트 엘리먼트에 q'' 의 결과 서브 트리들(즉, q'' 의 대상 하부 문서 각각으로부터 검색된 서브 트리들)을 추가로 묶어주면 된다(2.1절의 실체부의 표현 형태에 관한 기술 참조).

3. 성능 분석

본 절에서는 XQL 질의 처리에 있어 질의 분할에 의해 XML 실체부를 이용한 질의 처리와 실체부를 사용하지 않는 질의 처리 간의 성능을 비교한다. 본 논문의 질의 처리 비용 모델은 다음과 같다. 질의 처리 비용은 XML 저장소를 대상으로 질의를 처리하는 과정에서 접근되는 XML 엘리먼트 수의 총합에 비례한다. 따라서 본 성능 분석에서는 질의 처리 과정에서 접근되는 XML 엘리먼트 수의 총합을 성능 척도로 하였다. 3.1절에서는 성능 분석에 사용되는 파라미터에 대해 설명한다. 3.2절에서는 성능 분석 결과를 기술한다.

3.1 성능 파라미터

성능 분석에 사용되는 파라미터는 표 1과 같다. q 는 XML 저장소를 대상으로 한 질의를 나타내는 것으로, 그림 1 (b)와 같이 실체부에 대한 질의 q' 과 하부 데이터에 대한 질의 q'' 으로 분할된다. R_{view} 는 질의 분할에 의해 생성된 q' 의 결과 엘리먼트 개수를 나타내고, R_{xml} 은 q'' 의 결과 엘리먼트 개수를 나타낸다. 따라서 질의 q 의 결과 엘리먼트 개수는 $(R_{view} + R_{xml})$ 이 된다. $C_{refresh}$ 는 질의 처리에 사용되는 XML 실체부의 갱신 비용으로 $u \times x \times t$ 로 나타낼 수 있다. 여기서 u 는 단위 시간 당 뷰의 하부 XML 문서에 대한 변경 회수를 나타내며, x 는 하부 XML 문서에 대한 한번의 변경을 뷰 갱신에 반영하기 위한 엘리먼트 접근 수이다. 따라서 $u \times x$ 는 단위 시간 당 뷰 갱신 비용이 된다. t 는 해당 XML 실체부 접근 후 다시 접근할 때까지의 시간 간격을 나타낸다(실체부 갱신 비용에 대한 자세한 분석은 [6]을 참조).

3.2 성능 분석 결과

XQL 질의 처리의 성능은, 먼저 XML 실체부를 이용한 질의 처리와 실체부를 사용하지 않는 질의 처리로 나누어 분석 후 이들을 서로 비교한다.

먼저 XML 실체부를 이용한 질의 처리의 경우, XQL 질의 처리 비용 R_{UV} 는 기본적으로 아래 (식 1)에서 열거한 네 가지 항들의 합으로 구성된다.

$$\begin{aligned}
 R_{UV} &= \text{실체부에 대한 질의 } q' \text{ 처리 비용(A)} \\
 &+ \text{하부 데이터에 대한 질의 } q'' \text{ 처리 비용(B)} \\
 &+ \text{결과 통합 비용(C)} \\
 &+ \text{실체부 갱신 비용(D)} \quad \text{(식 1)}
 \end{aligned}$$

첫 번째 항(이하 A항)은 질의 분할에 의해 생성되는 q' 을

처리하는 비용이다. 두 번째 항(이하, B항)은 A항과 마찬가지로 질의 분할에 의해 생성되는 q'' 을 처리하는 비용이다.

표 1 성능 파라미터

파라미터 이름	내용
q	하부 XML 문서를 대상으로 한 질의
q'	XML 실체부를 대상으로 한 질의
q''	XML 실체부에 포함되지 않은 내용을 직접 하부 XML 문서로부터 검색하는 질의
DN_{xml}	하부 데이터에 저장된 XML 문서 중에서 같은 DTD를 참조하는 XML 문서 수
EN_{xml}	같은 DTD를 참조하는 XML 문서의 평균 엘리먼트 개수
H_{xml}	하부 XML 문서 접근을 위한 인덱스 접근 비용
R_{view}	XML 실체부를 구성하는 엘리먼트의 개수 (즉, q' 결과의 엘리먼트 개수)
R_{xml}	실체부를 이용한 질의 처리 시 하부 XML 문서에서 검색되는 엘리먼트의 개수 (즉, q'' 결과의 엘리먼트 개수)
N_{view}	q' 결과를 위한 검색 대상 XML 문서의 수
N_{xml}	q'' 결과를 위한 검색 대상 XML 문서의 수
$C_{refresh}$	XML 실체부 갱신 비용
$P_{q''}$	q'' 의 조건 엘리먼트(들)이 모두 인덱스를 가질 비율
P_q	q 의 조건 엘리먼트(들)이 모두 인덱스를 가질 비율
α	인덱스 접근 비용의 엘리먼트 접근 회수로의 환산율
β	실체부 갱신 비용의 엘리먼트 접근 회수로의 환산율

세 번째 항(이하, C항)은 q' 의 결과와 q'' 의 결과를 하나의 결과로 통합하는데 소요되는 비용이다. 네 번째 항(이하, D항)은 해당 실체부를 도출한 하부 XML 문서에 발생한 변경을 실체부에 반영하는데 드는 비용이다.

먼저, A항 즉, 실체부에 대한 질의 q' 을 처리하는 비용 $R_{q'}$ 은 (식 2)와 같다.

$$R_{q'} = R_{view} \quad \text{(식 2)}$$

(식 1)의 B항 즉, 하부 데이터에 대한 질의 q'' 을 처리하는 비용 $R_{q''}$ 은 (식 3)과 같다. 인덱스를 이용할 경우, 인덱스 접근 비용과 DTD 테이블의 첫 번째 엘리먼트 접근, 그리고 q'' 결과를 위한 검색 대상 XML 문서(들)의 엘리먼트 접근 비용의 합으로 구성된다. 인덱스를 이용하지 않을 경우, 전체 하부 XML 문서 접근 비용과 q'' 결과를 위한 검색 대상 XML 문서(들)의 엘리먼트 접근 비용의 합으로 구성된다.

$$\begin{aligned}
 R_{q''} &= P_{q''} \times (\alpha \times H_{xml} + 1 + N_{xml} \times EN_{xml}) \\
 &+ (1 - P_{q''}) \times (T_{xml} + N_{xml} \times EN_{xml}) \quad \text{(식 3)}
 \end{aligned}$$

이 때, T_{xml} 은 하부 데이터에 저장된 XML 문서 중에서 같은 DTD를 참조하는 문서들의 엘리먼트 수의 합을 의미하며 (식 4)로 나타낸다.

$$T_{xml} = DN_{xml} \times EN_{xml} \quad (\text{식 4})$$

(식 1)의 C항 즉, q' 과 q'' 을 처리한 결과를 통합하는 비용 R_I 는 (식 5)와 같다. 비용 R_I 는 질의 q 의 검색 결과 $R_{view} + R_{xml}$ 을 디스크에 저장하고, 이를 읽어 통합하는 디스크 I/O 합으로 구성된다.

$$R_I = 2 \times (R_{view} + R_{xml}) \quad (\text{식 5})$$

마지막으로 (식 1)의 D항은 $\beta \times C_{refresh}$ 로 나타낸다.

(식 2), (식 3)과 (식 5)를 (식 1)에 대입하여 정리하면 (식 6)과 같다.

$$\begin{aligned} R_{UV} &= R_{q'} + R_{q''} + R_I + C_{refresh} \\ &= 3 \times R_{view} + 2 \times R_{xml} + T_{xml} + N_{xml} \times EN_{xml} \\ &\quad + P_{q'} \times (\alpha \times H_{xml} + 1 - T_{xml}) + \beta \times C_{refresh} \end{aligned} \quad (\text{식 6})$$

다음은 실체뷰를 이용하지 않고 질의를 처리의 경우로서, XQL 질의는 하부 데이터의 XML 문서에 대한 접근을 통해 처리된다. 이를 고려하여 XQL 질의 처리 비용 R_{NV} 를 나타내면 (식 7)과 같다.

$$\begin{aligned} R_{NV} &= P_q \times (\alpha \times H_{xml} + 1 + (N_{view} + N_{xml}) \times EN_{xml}) \\ &\quad + (1 - P_q) \times (T_{xml} + (N_{view} + N_{xml}) \times EN_{xml}) \\ &= T_{xml} + (N_{view} + N_{xml}) \times EN_{xml} + P_q \times (\alpha \times H_{xml} + 1 - T_{xml}) \end{aligned} \quad (\text{식 7})$$

(식 8)은 XML 실체뷰를 이용하는 방법과 이용하지 않는 방법의 질의 처리 디스크 I/O 회수를 비교하기 위해 (식 7)에 (식 6)을 뺀 식 $R_{NV} - R_{UV}$ 를 나타낸 것이다.

$$\begin{aligned} R_{NV} - R_{UV} &= P_q \times (\alpha \times H_{xml} + 1 - T_{xml}) - P_{q'} \times (\alpha \times H_{xml} + 1 - T_{xml}) \\ &\quad + N_{view} \times EN_{xml} - 3 \times R_{view} - 2 \times R_{xml} - \beta \times C_{refresh} \end{aligned} \quad (\text{식 8})$$

(식 8)의 간략화를 위해 $(\alpha \times H_{xml} + 1 - T_{xml})$ 과 $(N_{view} \times EN_{xml} - 3 \times R_{view} - 2 \times R_{xml})$ 을 각각 X 와 Y 로 대치하면 (식 9)와 같다.

$$R_{NV} - R_{UV} = X \times (P_q - P_{q'}) + Y - \beta \times C_{refresh} \quad (\text{식 9})$$

XML 실체뷰를 이용하여 질의를 처리하는 방법의 성능이 실체뷰를 이용하지 않고 처리하는 방법보다 더 우수하기 위

해서는, R_{NV} 가 R_{UV} 보다 커야 한다. 즉,

$$R_{NV} - R_{UV} > 0 \quad (\text{식 10})$$

가 되어야 한다.

질의 q 가 주어지면, (식 9)에서의 X 와 Y , $P_{q'}$ 과 P_q , 그리고 $C_{refresh}$ 의 값들은 XML 저장소 내 메타 데이터로부터 추정할 수 있다. 따라서 이들 값들을 이용하여 질의 q 에 대한 (식 9)의 성립 여부를 판단할 수 있다. 즉, 질의 분할이 가능한 XQL 질의가 주어졌을 때, 질의 처리기는 질의 분할 처리 방법을 사용하는 것이 더 효율적인지 아니면 실체뷰를 이용하지 않는 질의 처리 방법을 사용하는 것이 더 효율적인지 판단할 수 있다.

(식 10)에서 하부 XML 문서의 인덱스 접근 비용(H_{xml}), 하부 데이터에 저장된 XML 문서 중에서 같은 DTD를 참조하는 문서들의 엘리먼트 수(T_{xml}), 질의 결과에 해당되는 엘리먼트의 수($R_{view} + R_{xml}$)가 일정하다면, 결국 $P_{q'}$ 과 P_q 의 값과 $C_{refresh}$ 에 의해 질의 분할 처리의 성능이 결정됨을 알 수 있다. 따라서 비율 $P_{q'}$ 과 P_q 의 차이가 클 수 있는 적절한 질의의 선택과 실체뷰 관리 비용 $C_{refresh}$ 를 줄일 수 있는 실체뷰의 효과적인 관리가 이루어진다면, XML 실체뷰의 사용은 XQL 질의 처리에 있어 아주 좋은 성능을 나타낼 것이다.

4. 결론

본 논문은 XML 저장소 내에 XML 실체뷰가 있다고 가정할 때, 주어진 질의를 (1) 실체뷰에 대한 질의와 하부 데이터에 대한 질의로 분할한 후 이들을 처리하여 그 결과를 통합하는 방법과 (2) 원래의 질의를 실체뷰를 이용하지 않고 처리하는 방법 간의 성능을 비교, 분석하였다. 분석 결과, 질의 분할 처리 기법에서 실체뷰 관리 비용이 적을수록, 그리고 상기 하부 문서들에 대한 질의 처리 부분에서 인덱스를 이용하여 질의를 처리할 비율이 높을수록, 실체뷰를 이용한 질의 처리가 실체뷰를 이용하지 않는 질의 처리보다 성능이 더 좋음을 확인할 수 있었다.

참고 문헌

- [1] S. Abiteboul, "On Views and XML," Proc. ACM Symp. on Principles of Database System, pp. 1-9, 1999.
- [2] 임재국 외, "점진적 갱신에 기반을 둔 XML 형성뷰 관리 프레임워크," 정보처리학회논문지 D, 제 8-D권 제 4호, pp. 327-338, 2001. 8.
- [3] 김수희 외, "XML 실체뷰를 이용한 XQL 질의 처리," 정보처리학회논문지 D, 제 8-D권 제 5호, pp. 461-472, 2001. 10.
- [4] J. Robie et al., "XML Query Language(XQL)," <http://www.w3.org/TandS/QL/QL98/pp/xql.html>, 1998.
- [5] 문찬호 외, "웹퍼 상에서의 XML 실체뷰를 이용한 XQL 질의 처리," 한국정보과학회 2001 가을 학술 발표 논문집(I), 제 27권 2호, pp. 6-8, 2001. 10.
- [6] 임재국 외, "XML 문서 실체 뷰의 점진적 갱신의 성능 분석," 한국 정보처리학회 2001 봄 학술발표 논문집, 8권 1호, pp. 19-22, 2001. 4.