

XML 실체 뷰를 이용한 XQL 질의 처리의 성능 평가

김수희^o, 문찬호, 강현철
 중앙대학교 컴퓨터공학과

e-mail:{shkim, moonch, hckang}@dmlab.cse.cau.ac.kr

Performance Evaluation of XQL Processing Using XML Materialized Views

SooHee Kim^o, ChanHo Moon, Hyunchul Kang
 Dept. of Computer Science and Engineering, Chung-Ang University

요약

XQL은 W3C에 의해 제안된 XML 질의 언어 중 하나이다. XML 문서 검색의 성능 향상을 위해서 XML 저장소에는 XML 문서들 외에 그들로부터 도출된 XML 뷰를 실체 뷰로 저장해 둘 수 있는데, 이들 실체 뷰는 XQL과 같은 XML 질의어로 명시된 질의 처리에 이용될 수 있다. 즉, XML 저장소에 대한 XQL 질의가 주어졌을 때 그 결과를 XML 실체 뷰로부터도 얻을 수 있다면 원래의 질의를 실체 뷰에 대한 질의로 변환하여 수행함으로써 XQL 질의 처리의 성능 향상을 가져올 수 있다. 본 논문에서는 XML 저장소 내에 XML 실체 뷰가 있다고 가정하고 이를 이용한 XQL 질의 처리의 성능을 XML 실체 뷰를 사용하지 않는 경우와 비교, 평가한다.

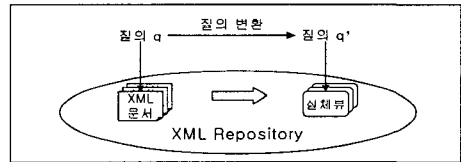
1. 서론

XQL(XML Query Language)은 1998년 W3C에 의해 제안된 XML(eXtensible Markup Language) 질의 언어 중 하나이다[1]. XQL은 경로 표현(Path Expression)을 이용하여 질의의 대상이 되는 입력 노드를 표시하고 그 의 조건을 표시하는 필터 연산자([]), 비교 연산자(=, !=, \$lt\$, \$gt\$,...)등을 이용하여 질의를 표현한다. 작성된 질의는 하나의 XML 문서 또는 XML 저장소(XML Repository)에 대하여 수행되어 그 결과로 하나의 루트 노드를 가진 잘 형성된(well-formed) XML 문서를 반환하게 된다.

<그림 1>에서 보는 것처럼, XML 저장소에는 하나의 DTD를 참조하는 여러 XML 문서들과 이들로부터 뷰 정의(View Definition)를 통해 도출된 XML 실체 뷰(Materialized View)[2]들이 있다고 하자. 이와 같은 XML 저장소에 대해 XQL 질의 q가 주어졌을 때 그 결과를 XML 실체 뷰로부터도 얻을 수 있다면 질의 변환을 통해 질의 q를 XML 실체 뷰에 대한 질의 q'으로 변환하고 XQL 질의 q의 결과와 동일한 결과를 XML 실체 뷰로부터 얻으면 XQL 질의 처리의 성능 향상을 가져올 수 있다[3]. 본 논문은 XML 저장소 내에 XML 실체 뷰가 있다고 가정하고 이를 이용한 XQL 질의 처리의 성능을 평가한 것이다.

또한, 성능 평가를 통해 XML 실체 뷰를 XQL 질의 처리에 사용하지 않는 경우에 비해 XML 실체 뷰를 XQL 질의 처리에 이용하는 것이 더 좋은 성능을 나타내기 위한 조건에 대해서도 기술한다.

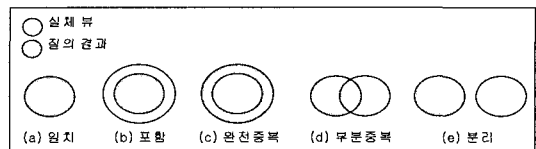
본 논문의 구성은 다음과 같다. 2절에서는 실체 뷰를 이용한 질의 처리의 유형에 대해 기술하고, 3절에서는 질의 변환을 지원하기 위한 XML 저장소의 구조를 설명한다. 4절에서는 성능 평가 결과를 기술하며, 5절에서 결론을 맺는다.



<그림 1> XML 실체 뷰를 이용한 XQL 질의 처리

2. 실체 뷰를 이용한 질의 처리의 유형

질의 처리에 실체 뷰를 이용하기 위해서는 주어진 질의 결과의 전부 또는 일부가 실체 뷰로부터 도출 가능해야 한다. 질의 결과와 실체 뷰 내용 간의 포함 관계는 <그림 2>와 같이 다섯 가지로 분류할 수 있다[4].

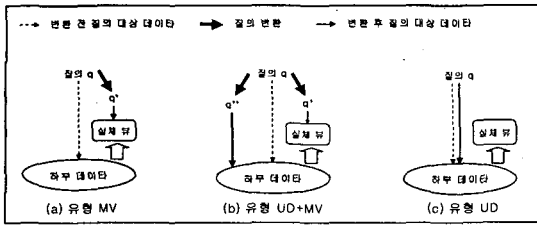


<그림 2> 실체 뷰와 질의 결과 간의 포함관계[4]

<그림 2>에서 (a)는 실체 뷰와 질의 결과가 일치하는 경우를 나타낸 것이며, (b)는 실체 뷰가 질의 결과를 포함하는 경우이다. (c)는 (b)와 반대로 실체 뷰가 질의 결과에 완전히 포함됨을 나타내며, (d)는 실체 뷰와 질의 결과가 일부만 겹침을 나타낸다. (e)의 경우에는 실체 뷰와 질의 결과 사이에 아무런 포함 관계가 성립하지 않는다.

<그림 2>와 같은 실체 뷰와 질의 결과 간의 포함 관계에 따라 질의 처리에 실체 뷰를 이용하는 방법은 <그림 3>과 같이 크게 세 가지 유형으로 나누어 볼 수 있다.

* 본 연구는 한국과학재단 목적 기초 연구(2000-1-30300-001-3) 지원으로 수행되었음.

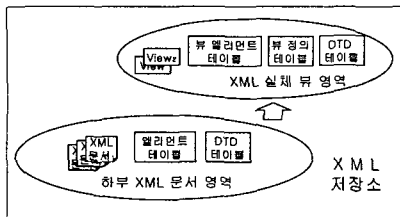


<그림 3> 실제 뷰를 이용한 질의 처리의 유형

<그림 3>에서 (a)의 유형 MV(Materialized View Only)는 <그림 2>의 (a)일치나 (b)포함에 해당하는 경우의 질의 처리 방법으로 주어진 질의 q 를 실제 뷰에 맞는 질의 q' 으로 변환하여 실제 뷰로부터 원하는 결과를 모두 얻을 수 있음을 나타낸다. (b)의 유형 UD+MV(Both Underlying Documents and Materialized View)는 <그림 2>의 (c)완전 중복이나 (d)부분 중복에 해당하는 경우의 질의 처리 방법으로 질의 결과의 일부는 실제 뷰에 존재하지만 일부는 실제 뷰에 존재하지 않는 경우이다. 따라서 유형 UD+MV는 주어진 질의 q 를 실제 뷰에 대한 질의 q' 과 하부 데이터에 대한 질의 q'' 으로 변환하여 각각으로부터 결과를 얻는다. (c)의 유형 UD(Underlying Documents Only)는 주어진 질의 결과를 모두 하부 데이터로부터 얻는 경우이다. <그림 2>의 (e)분리의 경우 이러한 질의 처리 유형을 사용하여야 하며, (b)포함의 경우에도 질의 처리 비용을 고려하여 유형 UD와 같은 질의 처리가 가능하다.

3. XML 저장소

본 논문의 XML 저장소 구조는 <그림 2>와 같이 크게 하부 XML 문서 영역(이하 하부 문서 영역)과 XML 실제 뷰 영역(이하 실제 뷰 영역)으로 나누어 볼 수 있다.



<그림 2> XML 저장소의 구성

하부 XML 문서 영역은 XML 문서들을 엘리먼트 단위로 나누어 저장한 엘리먼트 테이블과 DTD 테이블로 구성되며, XML 실제 뷰 영역 또한 XML 실제 뷰 정의로부터 얻어진 XML 실제 뷰들을 엘리먼트 단위로 나누어 저장한 뷰 엘리먼트 테이블과, 뷰 정의를 저장하는 뷰 정의 테이블, DTD 테이블로 구성된다.

4. 성능 평가

본 절에서는 XQL 질의 처리에 있어 XML 저장소에서 XML 실제 뷰가 지원되는 경우와 지원되지 않는 경우의 질의 처리 성능을 비교하며 XQL 질의 처리에 있어 XML 실제 뷰가 지원되는 경우가 지원되지 않는 경우보다 더 좋은 성능을 나타내기 위한 조건을 구한다. 또한 XML 실제 뷰가 지원되는 경우 주어진 파라미터 값의 변화가 질의 처리 성능에 미치는 영향에 대해서도 평가한다.

4.1절에서는 성능 척도에 대해 기술하며 4.2절에서는 성능 평가에 사용되는 파라미터에 대해 설명한다. 4.3절에서는 4.2절에서 주어진 파라미터를 이용한 성능 평가 결과를 기술하고, 4.4절에서는 성능 비교의 예를 기술한다.

4.1. 성능 척도

3절에서 기술한 XML 저장소는 DTD 테이블, 엘리먼트 테이블, 뷰 정의 테이블, 뷰 엘리먼트 테이블로 구성되고 XQL 질의 처리 시 이들 테이블에 접근하게 된다. 본 논문에서는 이러한 XML 저장소 구조를 고려하여 XQL 질의 처리의 성능 척도를 디스크 I/O 회수에 영향을 주는 상기 테이블들의 레코드 접근 수로 한다.

4.2. 성능 파라미터

성능 평가에 사용되는 파라미터는 <표 1>과 같다.

<표 1> 성능 파라미터

파라미터 이름	내용
EN_{DTD}	DTD 테이블의 레코드 수
DN_{view}	뷰의 개수
$EN_{view(i)}$	각 뷰의 엘리먼트 개수 ($i = 1, 2, \dots, DN_{view}$)
DN_{xml}	XML 저장소의 총 XML 문서 수
$EN_{xml(i)}$	각 문서의 엘리먼트 개수 ($i = 1, 2, \dots, DN_{xml}$)
p	관련 XML 실제 뷰에서 결과를 모두 얻을 수 있는 질의가 들어올 확률
$C_{refresh}$	XML 실제 뷰 갱신 비용

EN_{DTD} 는 한 DTD를 구성하는 레코드 수로 DTD의 엘리먼트 개수를 의미한다. DN_{view} 는 뷰의 개수로 뷰 정의 테이블의 레코드 수에 해당하며 $EN_{view(i)}$ 는 뷰 V_i 의 엘리먼트 개수를 의미한다. $EN_{xml(i)}$ 는 XML 문서 D_i 의 엘리먼트 개수이고 DN_{xml} 은 XML 저장소에 저장된 하부 XML 문서의 개수로, 엘리먼트 테이블의 총 레코드 수는 $EN_{xml(1)}$ 부터 $EN_{xml(DN_{xml})}$ 까지의 합으로 계산된다. p 는 <그림 2>의 (a)일치나 (b)완전 중복과 같이 XQL 질의 결과를 모두 관련 XML 실제 뷰에서 얻을 수 있는 질의가 들어올 확률이다. 즉, p 는 유형 MV로 처리가 가능한 질의가 들어올 확률을 의미한다. $C_{refresh}$ 는 질의 처리에 사용되는 XML 실제 뷰의 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 반영하는 비용이다. 따라서 $u \times x$ 는 단위 시간 당 뷰 갱신 비용이 된다. t 는 해당 XML 실제 뷰 접근 후 다시 접근할 때까지의 시간 간격을 나타낸다(실체 뷰 갱신 비용에 대한 자세한 분석은 [2]를 참조).

4.3. 성능 분석

XQL 질의 처리의 성능은, 먼저 XML 저장소에서 XML 실제 뷰가 지원되지 않는 경우와 지원되는 경우로 나누어 분석 후 이들을 서로 비교한다.

XML 실제 뷰가 지원되지 않는 경우, XQL 질의는 모두 하부 문서 영역의 엘리먼트 테이블의 레코드 접근을 통해 처리된다. 따라서 XML 실제 뷰가 지원되지 않는 경우, XQL 질의를 처리할 때 필요한 레코드 접근 수 R_{xml} 은

$$R_{xmi} = EN_{xmi(1)} + EN_{xmi(2)} + \dots + EN_{xmi(DN_{xmi})} = \sum_{i=1}^{DN_{xmi}} EN_{xmi(i)}$$

와 같이 계산된다. 그런데 XML 문서의 평균 엘리먼트 수, $avg(EN_{xmi})$ 는

$$avg(EN_{xmi}) = \frac{\sum_{i=1}^{DN_{xmi}} EN_{xmi(i)}}{DN_{xmi}}$$

와 같이 구할 수 있으므로 R_{xmi} 은 (식1)과 같이 나타낼 수 있다.

$$R_{xmi} = DN_{xmi} \times avg(EN_{xmi}) \quad (식 1)$$

한편, XML 실제 뷰가 지원되는 경우에 XQL 질의 처리는 크게 세 단계를 거쳐 수행된다. 첫 단계에서는 DTD 테이블과 뷰 정의 테이블에 접근하여 <그림 3>의 질의 처리 유형을 결정한다. 이에 필요한 레코드 접근 수는 $EN_{DTD} + DN_{view}$ 와 같다. 한편, 본 논문에서는 실제 뷰 갱신 방법으로는 뷰가 요청되었을 때 점진적으로 갱신을 수행하는 지연 갱신을 사용한다고 가정한다. 따라서 두 번째 단계에서는 결정된 질의 처리 유형이 MV 또는 MV+UD인 경우에 하부 데이터에 대한 변경을 관련된 실제 뷰에 반영하는 실제 뷰 갱신 과정을 거치게 되며 이 비용은 $C_{refresh}$ 가 된다. 마지막으로 세 번째 단계에서는 질의 처리에 필요한 테이블에 접근하게 되는데 이때 질의 처리 유형 x ($x \in \{MV, MV+UD, UD\}$)에 의한 질의 처리에 필요한 테이블 레코드 접근 수를 $r_{view(x)}$, 유형 MV로 처리가 가능한 질의가 들어올 확률을 $p_{mv}(=p)$, 유형 MV+UD로 질의 처리가 가능한 질의가 들어올 확률을 p_{ud} 라 하면, XML 실제 뷰가 지원되는 경우, XQL 질의를 처리 할 때 필요한 레코드 접근 수 R_{view} 는 (식 2-a)와 같이 나타낼 수 있다.

$$R_{view} = (EN_{DTD} + DN_{view}) + p_{mv} \times (r_{view(MV)} + C_{refresh}) + p_{ud} \times (r_{view(MV+UD)} + C_{refresh}) + p_{ud} \times r_{view(UD)} \quad (식 2-a)$$

먼저 $r_{view(MV)}$ 는

$$r_{view(MV)} = \lceil \log_n DN_{view} \rceil + EN_{view(i)} \quad (식 2-b)$$

와 같고, $r_{view(MV+UD)}$ 는 MV에서와 동일한 레코드 접근 수에 XML 실제 뷰에서 얻지 못한 나머지 결과를 얻기 위한 엘리먼트 테이블의 레코드 수가 더해지므로

$$r_{view(UD+MV)} = \lceil \log_n DN_{view} \rceil + EN_{view(i)} + R_{xmi} \quad (식 2-c)$$

이 된다. $r_{view(UD)}$ 는 엘리먼트 테이블의 레코드 수가 되므로,

$$r_{view(UD)} = R_{xmi} \quad (식 2-d)$$

와 같다.

그런데 (식 2-c)와 같이, XQL 질의 결과의 일부만 XML 실제 뷰에 포함된 경우에는, 연관된 XML 실제 뷰로부터 결과의 일부를 얻고 나머지 결과를 얻기 위해 다시 엘리먼트 테이블에 접근해야 한다. 따라서, 엘리먼트 테이블에 대한 별도의 인덱스를 고려하지 않을 경우, 관련 XML 실제 뷰에 접근하는 데

필요한 레코드 수에 엘리먼트 테이블 접근에 따라 R_{xmi} 만큼의 레코드 접근 수가 더해지므로, 결과적으로 (식 2-d)와 같이 질의 결과를 모두 엘리먼트 테이블로 얻는 경우보다 좋지 않은 성능을 내게된다. 따라서 엘리먼트 테이블에 대한 인덱스가 존재하지 않고 XQL 질의 결과의 일부만 XML 실제 뷰에 있을 경우에는 MV+UD 질의 처리 유형을 채택하지 않고 XQL 질의 결과를 모두 엘리먼트 테이블로부터 구하는 UD 질의 처리 유형을 사용하는 것이 효율적이다.

따라서 (식 2-a)에 (식 2-b)와 (식 2-d)를 적용하고 결과를 모두 XML 실제 뷰로부터 얻을 수 있는 질의가 들어올 확률 p 즉, p_{mv} 를 사용하여 R_{view} 을 다음과 같이 구한다.

$$R_{view} = (EN_{DTD} + DN_{view}) + p \times (\lceil \log_k DN_{view} \rceil + \frac{\sum_{i=1}^{DN_{xmi}} EN_{view(i)}}{DN_{view}} + C_{refresh}) + (1-p) \times R_{xmi} \quad (식 2-e)$$

(식 2-e)에서 XML 실제 뷰의 평균 엘리먼트 수를, $avg(EN_{view})$ 라 하면 결국 R_{view} 는

$$R_{view} = (EN_{DTD} + DN_{view}) + p \times (\lceil \log_k DN_{view} \rceil + avg(EN_{view}) + C_{refresh}) + (1-p) \times R_{xmi} \quad (식 2)$$

이 된다.

또한 XML 실제 뷰가 지원되는 경우의 XQL 질의 처리가 지원되지 않는 경우의 질의 처리보다 더 우수한 성능을 나타내기 위해서는, R_{xmi} 이 R_{view} 보다 커야 한다. 즉,

$$R_{xmi} - R_{view} > 0 \quad (식 3-a)$$

이 되며 이에 (식 2)를 대입하면,

$$R_{xmi} - [(EN_{DTD} + DN_{view}) + p \times (\lceil \log_k DN_{view} \rceil + avg(EN_{view}) + C_{refresh}) + (1-p) \times R_{xmi}] > 0$$

$$p \times (R_{xmi} - \lceil \log_k DN_{view} \rceil - avg(EN_{view}) - C_{refresh}) - (EN_{DTD} + DN_{view}) > 0 \quad (식 3)$$

이 된다. 즉 XQL 질의가 주어졌을 때 (식 3)이 성립되면 XML 실제 뷰가 지원되는 경우의 질의 처리가 지원되지 않는 경우의 질의 처리보다 우수한 성능을 나타내게 된다. 만약 XML 저장소 내에 XML 문서의 수와 DTD의 엘리먼트 수, 실제 뷰의 개수가 일정하다면, 결국 p 의 값과 $C_{refresh}$ 에 의해 질의 처리의 성능이 결정됨을 알 수 있다. 따라서 확률 p 를 높일 수 있는 적절한 실제 뷰의 선택과 $C_{refresh}$ 를 줄일 수 있는 효율적인 뷰 관리가 실제 뷰를 이용한 질의 처리에 가장 중요한 요소가 된다.

(식 3-a)가 만족되기 위한 확률 p 의 최소값을 구하면 (식 4)와 같다.

$$p > \frac{EN_{DTD} + DN_{view}}{R_{xmi} - \lceil \log_k DN_{view} \rceil - avg(EN_{view}) - C_{refresh}} \quad (식 4)$$

결국 XML 실체 뷰가 지원되고 이를 질의 처리에 이용할 경우, p 가 (식 4)와 같은 조건을 만족한다면 XQL 질의 처리에 있어 XML 실체 뷰가 지원되지 않는 경우에 비해 더 나은 질의 처리 성능을 가져오게 된다.

또한 (식 3-a)가 만족되기 위한 $C_{refresh}$ 의 최대값을 구하면 (식 5)와 같이 정리할 수 있다.

$$C_{refresh} < \frac{(R_{xml} - \lceil \log_2 DN_{view} \rceil - avg(EN_{view})) \cdot (EN_{DTD} + DN_{view})}{p} \quad (식 5)$$

한편, 확률 p 의 값이 0이라면 R_{view} 는 (식 2)로부터

$$R_{view} = (EN_{DTD} + DN_{view}) + R_{xml}$$

이 된다. 따라서, 이 경우 R_{xml} 보다 $EN_{DTD} + DN_{view}$ 만큼의 레코드 접근 수가 더 필요하다. 이는 실체 뷰 이용 시 주어진 질의의 처리 유형을 결정하기 위해 DTD 테이블과 뷰 정의의 테이블에 접근하게 됨으로써 생기는 레코드 접근 수 만큼의 오버헤드이다. 그러나 $EN_{DTD} + DN_{view} \ll R_{xml}$ 이므로 확률 p 의 값이 0이더라도 실체 뷰 이용에 따른 오버헤드는 R_{xml} 에 비해 크지 않음을 알 수 있다.

XQL 질의 처리시 접근하는 테이블 레코드 수를 통한 성능 평가의 결과, XML 실체 뷰 이용이 실체 뷰를 이용하지 않는 경우보다 더 좋은 성능을 나타내기 위해서는 XML 실체 뷰 관리 비용 $C_{refresh}$ 와 질의의 결과가 XML 실체 뷰에 포함되는 질의가 주어질 확률 p 가 큰 영향을 줄을 알 수 있다. 확률 p 가 낮을 경우엔 XML 저장소에서 XML 실체 뷰가 지원되지 않는 경우의 질의 처리가 오히려 실체 뷰가 지원되는 경우의 질의 처리보다 더 좋은 성능을 나타낼 수도 있다. 하지만, 최악의 경우 확률 p 의 값이 0이더라도 XML 실체 뷰가 지원되는 경우 필요한 레코드 접근 수와 지원하지 않는 경우 필요한 레코드 접근 수의 차이는 최대 $EN_{DTD} + DN_{view}$ 를 넘지 않는다.

따라서 확률 p 를 높일 수 있는 적절한 실체 뷰의 선택과 실체 뷰 관리 비용 $C_{refresh}$ 를 줄일 수 있는 실체 뷰의 효과적인 관리가 이루어진다면, XML 실체 뷰의 사용은 XQL 질의 처리에 있어 아주 좋은 성능을 나타낼 것이다.

4.4 성능 비교 예

본 절에서는 4.3절에서 얻어진 식들을 이용해 성능 비교의 예를 기술한다. 주어진 파라미터를 이용하여 XML 실체 뷰가 지원되는 경우와 지원되지 않는 경우, 질의 처리 시 필요한 레코드 접근 수를 비교한다.

DTD 테이블의 엘리먼트 개수가 20개, 동일한 DTD를 만족하는 XML 문서의 개수가 10개, XML 문서의 평균 엘리먼트 수가 50개이고, XML 실체 뷰의 개수가 3개, XML 실체 뷰의 평균 엘리먼트 수가 10개라고 가정하자. 즉, <표 1>의 파라미터 값은 다음과 같다.

EN_{DTD}	DN_{xml}	$avg(EN_{xml})$	DN_{view}	$avg(EN_{view})$
20	10	50	3	10

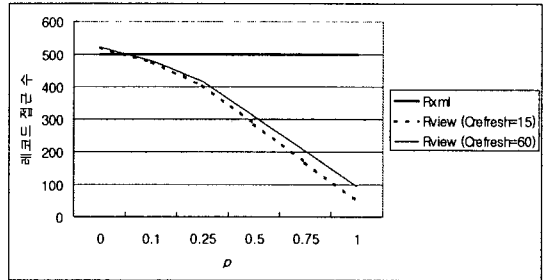
이들 파라미터 값에 따라 R_{xml} 은

$$R_{xml} = DN_{xml} \times avg(EN_{xml}) = 10 \times 50 = 500$$

이 된다. 반면 R_{view} 는

$$\begin{aligned} R_{view} &= (EN_{DTD} + DN_{view}) + p \times (\lceil \log_2 DN_{view} \rceil \\ &\quad + avg(EN_{view}) + C_{refresh}) + (1-p) \times R_{xml} \\ &= 20 + 3 + p \times (1 + 10 + C_{refresh}) + (1-p) \times 500 \\ &= 523 - p \times (489 - C_{refresh}) \end{aligned}$$

이 된다. 따라서 주어진 XQL 질의 결과가 XML 실체 뷰에 모두 포함될 확률 p 의 값에 따라, 질의 처리에 필요한 레코드 접근 수는 다음 그래프와 같이 나타낼 수 있다.



그래프에 나타난 것과 같이 확률 p 가 커짐에 따라, XML 실체 뷰가 지원되는 경우와 지원되지 않는 경우의 레코드 접근 수의 차이는 더 커졌다. 확률 p 가 0인 경우, R_{view} 은 R_{xml} 보다 크지만 그 차이가 미미하게 나타났으며, $C_{refresh}$ 이 커질수록 질의 처리에 필요한 레코드 접근 수가 약간씩 증가하였다.

5. 결론

본 논문은 XML 저장소에 대한 XQL 질의 처리의 성능 향상을 위해 XML 실체 뷰를 이용할 경우, 그 성능을 평가하였다. 성능 평가는 XML 저장소의 구조에서 XML 실체 뷰가 지원되는 경우와 지원되지 않는 경우, 필요한 테이블 접근 레코드 수의 비교를 통해 수행하였다. 성능 평가 결과, 관련 XML 실체 뷰에서 결과를 모두 얻을 수 있는 질의가 들어올 확률이 커질수록 XML 실체 뷰를 지원하는 경우의 성능이 더 좋아짐을 확인할 수 있었다.

참고 문헌

- [1] J. Robie et al., "XML Query Language (XQL)," <http://www.w3.org/TandS/QL/QL98/pp/xql.html>, 1998.
- [2] 임재국, "XML 문서의 실체 뷰를 위한 점진적 갱신," 석사 학위 논문, 중앙대학교 컴퓨터공학과, 2001. 2.
- [3] 김수희, "XML 실체 뷰를 이용한 XQL 질의 변환," 석사 학위 논문, 중앙대학교 컴퓨터공학과, 2001. 2.
- [4] D. Lee and W. Chu, "A Semantic Caching Scheme for Wrappers in Web Databases," Tech. Rep. TR-990004, UCLA, Feb. 1999.