분산 환경에 질의 재구성 기반의 XQuery 질의 최적화

박종현○ 최성일 강지훈
충남대학교 컴퓨터전공
{jonghyunpark○, sichoi, jhkang}@cnu.ac.kr

XQuery query Optimization Based on Query Rewriting over Distributed Environment

Jong-Hyun Park○, Seong-Il Choi, Ji-Hoon Kang
Dept. of Computer Science and Engineering, Chungnam National University

XML 데이터가 인터넷 상의 데이터 표현 및 교환을 위한 표준으로 자리 잡으면서 이를 효율적으로 검색하고 통합하기 위하여 W3C는 XQuery를 표준으로 제안하고 있다. XQuery에 대한 관심이 증가함에 따라 기존의 다른 질의어들과 마찬가지로 여러 응용에서 이를 효율적으로 처리하고자 하는 연구가 진행되었으며, 그 중 대표적인 한 분야가 XQuery 질의 최적화이다.

분산 환경에서 다수의 지역 시스템들에 저장된 XML 데이터들을 통합 하고 검색하기 위한 방법으로 XQuery 질의를 사용하는 것은 상호운용성 측면에서 자연스러운 선택이다. 또한 다수의 지역 시스템들을 기반으로 작성된 통합 XQuery 질의를 처리하기 위해서 이를 각 지역 시스템에 맞는 지역 XQuery 질의로 분할하여 처리하는 방법은 통합 SQL 처리와 같은 기존 통합 질의 처리를 위해서 널리 알려진 방법들 가운데 하나이다. 본 논문에서는 질의 분할을 기반으로 통합 XQuery 질의를 처리하는 Mediator에서 분할된 지역 XQuery 질의들을 각 지역 시스템에 질의하기 이전에 이를 최적화하는 방법을 제안한다. 물론 각 지역 시스템들은 내부적으로 질의 처리를 위한 최적화를 수행할 수 있다. 그러나 우리는 지역 시스템의 내부적인 최적화 방법과 무관하게 지역 시스템에 질의하기 위한 지역 질의 자체를 재작성하여 최적화한다. 즉, 본 논문의 최적화 목적은 지역 질의를 재작성하여 최적화된 지역 질의를 지역 시스템에 제공하는 것이 그 목적이다. XQuery 질의를 최적화하기 위한 앞선 몇몇 연구에서는 XML 스키마 또는 DTD와 같은 미리 정의된 XML 데이터의 구조정보를 이용하여 최적화하는 방법을 제안하고 있다. 그러나 현재 모든 응용이 XML 데이터를 위한 구조정보를 포함하고 있지는 않은 것이 현실이다. 그러므로 본 논문에서는 XQuery 질의의 특성을 파악하고 XQuery 질의 자체만을 이용한 최적화 방법을 제안한다.

본 논문에서는 XQuery 질의 최적화를 위하여 XQuery 질의의 특성들을 고려한 세 가지 질의를 최적화 방법을 제안한다. 첫 번째 방법은 XQuery 질의에 존재하는 불필요한 표현들을 제거하는 것이고, 두 번째 방법은 질의 재배치를 이용한 최적화 방법이다. 마지막으로 세 번째 방법은 XQuery가 For절에 의해서 중첩된다는 점을 고려하여 For절을 기반으로 불필요한 반복을 최소화 하여 최적화 방법이다.

Simplified Original XQuery query	Let clause Removement	XPath Exp Removement	Redundant condition Removement	Operation Replacement	Let clause Replacement
E1(FOR a in E2 FOR z in E3 **LET c := E4(a)** LET d := E5 LET e := E6(a) WHERE z/text()=e/text() and E5/@id ="1" and d/@id < "3" RETURN E7(a, E5, e))	E1(FOR a in E2 FOR z in E3 LET **d := E5** LET e := E6(a) WHERE z/text()="e/text()" and **E5**/@id ="1" and d/@id < "3" RETURN E7(a, **E5**, e))	E1(FOR a in E2 FOR z in E3 LET d := E5 LET e := E6(a) WHERE z/text()="e/text() **and d/@id = "1"** **and d/@id < "3"** RETURN E7(a, d, e))	E1(FOR a in E2 FOR z in E3 LET d := E5 LET e := E6(a) WHERE **z/text()="e/text()** **and d/@id = "1"** RETURN E7(a, d, e))	E1(FOR a in E2 FOR z in E3 LET d := E5 **LET e := E6(a)** WHERE d/@id = "1" and z/text()="e/text() RETURN E7(a, d, e))	E1(FOR a in E2 FOR z in E3 LET d := E5 WHERE d/@id = "1" and z/text()="e/text() RETURN E7(a, d, e))

그림 1. 불필요한 표현의 제거와 재배치를 이용한 최적화 방법

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기반기술개발사업의 지원에 의한 것임

XQuery 질의 내부에 불필요한 표현을 제거하기 위한 방법은 “사용하지 않는 LET절의 제거”, “중복된 XPath 표현의 제거”, “WHERE절의 조건 중 동치 또는 포함관계의 연산 중 의미적으로 중복된 부분을 제거”하는 세 가지 방법으로 구성되며, 그림 1은 각 경우의 예를 보인다. 또한 질의 재배치를 이용한 최적화 방법은 크게 연산자의 위치를 이동하는 방법과 LET절의 위치를 이동하는 방법으로 구분된다. 연산자의 위치를 이동하는 방법은 XQuery 질의의 WHERE절에 선언된 연산자들 중 Constant 값을 얻을 수 있는 연산자는 Constant 값으로 대체하고, Constant 값을 피연산자로 갖는 연산을 연산자들 중 가장 앞쪽으로 배치한다. 물론 XQuery 처리기가 XQuery 질의를 처리할 때, 연산자들 중 가장 앞쪽에 위치한 연산자를 가장 먼저 처리하지 않을 수도 있다. 그러나 대부분 XQuery 질의 처리기가 XQuery 질의를 처리하는 순서는 XQuery 구문 분석 트리로부터 bottom-up, left-right로 처리하므로 우리는 Constant 값을 갖는 연산자의 순서를 앞쪽으로 재배치하여 최적화를 수행한다. 그림에서 E는 XQuery의 Expression을 표현하고 C는 Condition을 표현한다.

For절 기반의 질의 재작성을 이용한 XQuery 질의 최적화의 목적은 질의 재작성에 의해서 불필요한 연산의 수행을 줄이는 것이다. 많은 경우 XQuery 질의의 반복은 FOR절로부터 야기되고 이러한 반복은 불필요한 중복 연산을 생성한다. 그러므로 우리는 어떤 형태의 FOR절이 불필요한 연산을 발생하는지 정의하고, 이를 줄이기 위한 질의 재작성 방법을 제안한다. FOR절 기반의 질의 재작성을 위해서 우리는 먼저 XQuery 질의에서 사용되는 FOR절의 위치에 따라 FOR절을 다음과 같이 크게 12가지로 구분한다.

XQ1-A	XQ1-B	XQ1-C	XQ2-A	XQ2-B	XQ2-C
For x in E1 For a in E2 Where C1(**x**) and C2(a) Return **x**	For a in E1 **For x in E2** For b in E3(a) Where C1(**x**) and C2(b, x) Return a, **x**	For a in E1 **For x in E2** Where C1(a) and C2(**x**) Return **x**	**For x in E1** For a in E2 Where C1(a) and C2(**x**) Return a	For a in E1 **For x in E2** For b in E3(a) Where C1(**x**) and C2(b) Return a	For a in E1 **For x in E2** Where C1(a) and C2(**x**) Return a
XQ3-A	**XQ3-B**	**XQ3-C**	**XQ4-A**	**XQ4-B**	**XQ4-C**
For x in E1 For a in E2 Where C1(a) Return a, **x**	For a in E1 **For x in E2** For b in E3(a) Where C1(b) Return a, **x**	For a in E1 **For x in E2** Where C1(a) Return a, **x**	**For x in E1** For a in E2 Where C1(a) Return a	For a in E1 **For x in E2** For b in E3(a) Where C1(b) Return a	For a in E1 **For x in E2** Where C1(a) Return a

그림 2. XQuery 질의에 존재하는 FOR절의 12가지 분류

For절 기반 질의 최적화의 기본 Idea는 Loop invariant를 이용하여 FOR절에 의해서 발생되는 반복 횟수를 줄이는 것이다. 이를 위하여 본 논문에서는 다음과 같은 3가지 질의의 재작성 방법을 이용한 XQuery 질의 최적화 방법을 제안한다. 첫 번째 “XQuery 질의의 Subset을 FOR절의 Inner Query로 재 작성하는 방법”은 위의 분류들 가운데 XQ1-A, XQ1-B, XQ1-C, XQ2-A, XQ2-B, 그리고 XQ2-C와 같은 경우의 질의를 위해서 적용된다. 두 번째 방법인 “XQuery 질의의 Subset을 LET절의 Inner Query로 재 작성하는 방법”은 그림 2의 XQ3-A와 XQ4-A 같은 경우의 질의최적화를 위해서 적용가능하다. 마지막으로 “XQuery 질의의 Subset을 RETURN절의 Inner Query로 재 작성하는 방법”은 XQ3-C와 XQ4-C와 같은 경우의 질의 최적화에 사용될 수 있는 방법이다. 그림 3은 FOR절 기반의 질의 최적화를 위한 우리의 세 가지 방법에 대한 몇 가지 예를 보인다.

Rewritten XQ1-A	Rewritten XQ1-C	Rewritten XQ2-B	Rewritten XQ2-C	Rewritten XQ3-A	Rewritten XQ4-C
For x' in **For x in E1** **Where C1(x)** **Return x** For a in E2 Where C2(a) Return **x'**	For a in E1 **For x' in** **For x in E2** **Where C2(x)** **Return x** Where C1(a) Return **x'**	For a in E1 **For x' in** **For x in E2** **Where C1(x)** **Return x** For b in E3(a) Where C2(b) Return a	For a in E1 **For x' in** **For x in E2** **Where C2(x)** **Return x** Where C1(a) Return a	Let a' := **For a in E2** **Where C1(a)** **Return a** **For x in E1** For a'' in a' Return a'', **x**	For a in E1 Where C1(a) Return **For x in E2** Return a

그림 3. XQuery 질의에 존재하는 FOR절의 12가지 분류

본 논문에서는 앞서 언급한 모든 방법들을 프로토타입으로 구현하여 각 방법의 성능을 평가하였다. 그 결과 우리의 최적화 알고리즘들은 지역 시스템에서 사용하는 XQuery 질의 처리기에 의존적으로 동작하기는 했지만, 평균적으로 질의의 처리시간을 줄일 수 있었다. 특별히 FOR절 기반 질의 재구성에 의한 질의 최적화 방법은 지역 시스템의 XQuery 질의 처리기에 무관하게 효율적인 성능을 보였다.

본 논문의 결과는 향후 분산 환경에 존재하는 데이터들을 XQuery 질의를 이용하여 통합하고 검색하기 위하여 효율적으로 사용될 것으로 기대된다. 또한 우리의 결과는 현재 매우 활발하게 연구되고 있는 단일 시스템에서 XQuery 질의를 효율적으로 처리하기위하나 연구와 상호 도움을 줄 수 있을 것으로 사료된다.