제26회 한국정보처리학회 추계학술발표대회 논문집 제13권 제2호 (2006. 11)

XML 데이터베이스를 위한 다차원 중포 엘리먼트 색인구조의 운용과 할당

이정아, 이종학
대구가톨릭대학교 컴퓨터정보통신공학부
e-mail:{leejunga, jhlee11}@cu.ac.kr

Operations And Assignments Of Multidimensional Nested Element Indexs For XML Databases

Jung-A Lee, Jong-Hak Lee
School of Computer and Information Communications Engineering, Catholic Univ. of Daegu

요 약

최근 XML 데이터베이스는 웹의 발전과 더불어 광범위한 인터넷의 자원 공유에 크게 기여하고 있다. XML로 작성된 문서를 저장하고 검색하기 위해 XML 문서의 저장, 질의언어, 질의처리 등에 대한 분야가 활발히 연구되고 있다. 특히 그 중 질의처리의 처리비용을 줄이기 위한 데이터 질의 최적화 기법에 관한 연구가 중요한 과제이다. 중포된 엘리먼트에 대한 기존의 색인기법들은 일차원 색인구조를 이용함으로써 XML Schema가 가지는 타입상속 개념을 고려한 XML 질의들에 대한 처리를 효율적으로 지원하지 못하는 문제점을 가지고 있다. 따라서 본 논문에서는 XML Schema가 가지는 타입상속 개념을 고려한 XML 질의들에 대한 처리를 효율적으로 지원할 수 있는 다차원 중포 엘리먼트 색인구조와 다차원 경로 엘리먼트 색인구조의 운용법을 제시한다. 또한 효과적인 질의 처리를 하기 위한 XML 데이터베이스 색인구조의 유지비용을 줄이기 위하여 저장 공간 및 갱신유지 비용을 최소화할 수 있는 효과적인 색인할당 방법을 제시한다.

1. 서론

XML(eXtensible Markup Language)[1]데이터베이스는 웹 데이터 표현 및 인터넷 문서 교환의 표준으로 채택되면서 XML 데이터들을 효율적으로 관리하기 위한 연구가 계속되고 있다. 그리고 XML 문서상의 구조적 제약 조건들은 DTD(Data Type Definition)와 XML Schema[2]에 의해 정의된다. XML 데이터베이스 안에 어떤 요소 형과 특성, 값들을 사용할 수 있는지에 대한 정의를 위해서 XML에서는 DTD를 널리 사용하고 있다. XML Schema는 표준 XML Schema로서 DTD보다 다양한 데이터 타입을 정의 할 수 있고, 속성, 정보 요소, 사용자 정의 데이터 타입에 대해서 상속을 지원하는 구조를 가지고 있다. XML은 반구조적 데이터(semi-structured data)의 일종으로 볼 수 있다. XML과 비정형 데이터는 유연한(flexible)구조를 손쉽게 표현할 수 있도록 그래프 형태의 데이터 모델[3]로 표현된다.

최근 XML로 작성된 문서를 저장하고 검색하기 위해 XML 문서의 저장, 질의언어, 질의처리 등에 대한 분야가 활발히 연구되고 있다. 특히 그 중 질의처리의 처리비용을 줄이기 위한 데이터 질의 최적화 기법에 관한 연구가 중요한 과제이다. 기존에 제안된 중포 엘리먼트(nested element)에 대한 색인은 B^+-tree와 같은 일차원 색인구조를 이용한다. 따라서, 질의의 대상범위가 타입상속 계층상의 임의의 타입들로 제한되거나, 질의에 포함된 복합 엘리먼트들의 타입이 타입상속 계층상의 임의의 타입들로 제한되는 경우에는 효율적으로 지원하지 못하는 문제점을 가진다. 즉, XML 데이터베이스가 가지는 타입상속 개념에 대한 고려를 하지 못하고 있다.

이와 같은 기존의 일차원 색인구조를 이용하는 중포 엘리먼트에 대한 색인기법들이 가지는 문제점을 해결하기 위하여, 타입상속 개념을 내포하고 있는 XML 질의를 효율적으로 처리하기 위한 다차원 색인구조를 이용한 다차원 중포 엘리먼트 색인기법[4]이 제안되었다. 본 논문에서는 다차원 중포 엘리먼트 색인구조에 대한 운용법을 제시한다. 또한 효과적인 질의처리를 하기위한 효율적인 색인할당방법을 제시한다.

본 논문의 구성은 다음과 같다. 먼저 제 2절에서는 관련연구로 XML 데이터의 구조적 정의인 DTD, XML Schema와 기존의 중포 엘리먼트에 대한 색인기법에 대해서 기술한다. 제 3절에서는 XML 데이터베이스의 중포 엘리먼트 색인기법으로 다차원 파일구조를 이용하는 다차원 중포 엘리먼트 색인구조를 소개하고 이들의 운영법들을 제시한다. 제 4절에서는 다차원 중포 엘리먼트 색인구조의 할당방법을 제시하고, 각 할당방법들에 대한 장·단점과 최적의 색인구조 할당방법을 제시한다. 마지막으로 제 5절에서는 결론을 기술한다.

2. 관련연구

XML(Extensible Markup Language)은 HTML을 대체

할 목적으로 W3C에서 표준화 작업을 진행하고 있는 페이지 기술 언어이며, HTML과 SGML(Standard Generailized Markup Language)의 장점을 모두 가지도록 규정한 구조화된 정보를 포함하고 있는 문서들을 위한 마크업 언어이다[1]. 즉, 문서의 내용과 형태를 다양하게 정의할 수 있는 강력한 기능을 가진 SGML에 기반을 두고 HTML의 단순함과 유연성을 고려하여 새롭게 만들어진 마크업 언어가 바로 XML이다. 그리고 XML은 문서 내용의 구조화와 문서를 보여주기 위한 스타일을 명백하게 구분함으로써 문서의 내용을 변경하지 않아도 다양한 형태를 갖는 문서를 만들어낼 수 있다.

DTD는 XML문서를 구성하는 정보 요소(element), 정보 요소의 구조와 속성등 문서의 형태를 구조화하여 정의한 것으로 XML의 정보 교환의 이점을 제공해 준다. 하지만 DTD는 정보 요소, 속성, 그리고 데이터 형 정의에 대한 상속의 매커니즘 등을 제공하지 않는 문제점을 지니고 있다. W3C에는 이를 개선하기 위해서 XML Schema[2]를 제안하였다.

XML Schema는 DTD보다 다양한 데이터 타입을 정의할 수 있고, 속성, 정보 요소, 사용자 정의 데이터 타입에 대해서 상속하는 구조를 정의할 수 있다. XML Schema는 기본적으로 type들로 구성되며 simple type과 complex type으로 나누어진다. simple type은 string, byte, integer, date를 비롯한 40가지 이상의 built-in 타입들을 제공하고 있으며, complex type은 속성, 엘리먼트를 포함한다.

XQuery[5]는 XML 문서에 대한 질의어로 W3C에서 제안한 새로운 표준안으로 XML과 별도의 문법체계를 가지며 FLWR(for, let, where, return) 절로 구성된다. for절은 SQL 질의의 from절과 의미상으로 유사하며, let절은 표현을 간략하게 하기 위해서 복잡한 식을 변수 이름에 배치할 수 있도록 한 것이다. where절은 SQL에서의 where절과 유사하며 단순 엘리먼트에 대한 조건인 단순 술어(simple predicate)와 더불어 중포 엘리먼트에 대한 조건인 중포 술어(nested predicate)를 사용할 수 있다.

XPath[6]는 XML 문서의 엘리먼트들을 노드(node)의 개념으로 접근하여 XML 문서의 색션들의 정확한 위치를 지정해 주기 위한 경로지정 문법으로 XML과 별도의 문법체계를 가진다. 또 XML 문서를 한 개의 트리(tree) 모델로 취급하며, 위치 경로에 있어서는 상대 위치경로와 절대 위치경로로 표현할 수 있다. XPath는 중포 엘리먼트의 경로를 표현하기 위한 경로식(path expression)이다. 본 논문에서도 이러한 경로식에서 경로상의 엘리먼트의 타입을 타입상속 계층상의 일부 타입들로 한정하여 표현할 수 있도록 XPath를 확장하여 이를 *확장된 XPath*라 한다. 확장된 XPath는 각 엘리먼트 다음에 타입의 이름이 올 수 있도록 확장한 것으로 다음과 같은 형태를 가진다. 단, E_i 뒤의 중괄호{ }는 선택적임을 나타내는 표시이다.

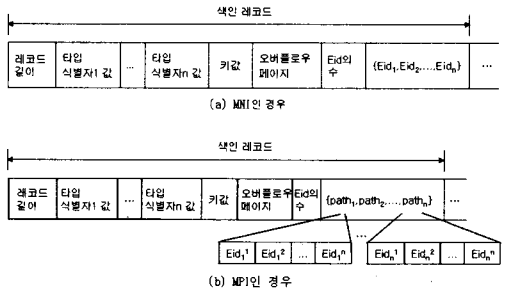
$$EP = T_1/E_1\{(T_2)\}/E_2\{(T_3)\}/\ldots/E_n\{(T_{n+1})\}$$

경로 EP에서 타입 T_1을 *타겟타입*, T_{i+1}을 엘리먼트 E_i의 *복합 엘리먼트 타입*이라 정의한다. 타겟 타입과 복합 엘리먼트 타입은 경로에서 타입상속 계층구조에 속하는 특정 타입으로 한정(limit)될 수 있으며, 이를 타입대치(type substitution)라 한다. 이러한 타입대치는 질의의 범위를 특정한 타입으로 한정할 수 있도록 하여 타입상속의 개념을 XML 질의에 표현하도록 한 것이다.

지금까지 제안된 기존의 중포 엘리먼트에 대한 색인기법으로는 DataGuide[7], 1-index[8], Index Fabric[9], APEX[10] 등이 있다. DataGuide는 비결정적(non-deterministic) 오토마타를 결정적(deterministic) 오토마타로 변환하는 과정과 동일한 과정으로 경로를 색인하는 기법이다. 일반적으로 비결정적 오토마타를 결정적 오토마타로 바꿀 경우, 크기가 커지게 되지만, XML 문서 내에 동일한 경로들이 많이 존재할수록 색인의 크기는 줄어든다. 1-index는 데이터 그래프 내의 각 노드들을 루트로부터 시작되는 경로의 집합이 동일한 노드들을 모아 색인을 구축하는 기법으로서, DataGuide와 마찬가지로 XML 문서 내에 동일한 경로가 매우 많이 존재한다는 점을 이용하는 색인기법이다. 그러나 이러한 색인구조들은 XML 데이터 모델의 타입상속의 특징을 반영하지 못하는 것들로서, 타겟타입의 대치 또는 복합 엘리먼트 타입의 대치가 있는 질의는 지원하지 못한다. 즉, 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, XPath식에 나타나는 어떠한 엘리먼트의 복합 엘리먼트 타입이 타입상속 계층상의 임의의 타입들로 제한이 되는 질의들을 지원할 수 없다.

3. 다차원 중포 엘리먼트 색인구조의 운용

본 논문에서는 XML 데이터베이스의 중포 엘리먼트에 대한 색인구조로 다차원 파일구조의 하나인 계층 그리드 파일[11]을 이용하여, 색인 엔트리를 구성하는 방법에 따라 다차원 중포 색인구조(multidimensinal nested index : MNI)와 다차원 경로 색인구조(multidimensinal path index : MPI)로 분류한다. MNI는 색인 엔트리를 색인된 중포 엘리먼트의 타겟타입 계층에 속하는 엘리먼트에 대한 엘리먼트 식별자(Eid)들로 구성하고, MPI는 색인 엔트리를 색인된 중포 엘리먼트에 대한 경로 인스턴스(Eid의 리스트)들로 구성한다. 그림 1은 다차원 중포 엘리먼트 색인구조의 색인 페이지 구조를 나타낸다. 그림 1(a)는 MNI의 색인 페이지 구조이며, 그림 1(b)는 MPI의 색인 페이지 구조이다.



(그림 1) 다차원 중포 엘리먼트 색인구조의 색인 페이지 구조

중포 엘리먼트에 대한 색인기법은 경로상의 참조관계에 의한 암시적 조인의 연산을 미리 계산하여 두는 것으로서 확장된 XPath식을 만족하는 엘리먼트의 탐색에는 매우 유용하지만, 상대적으로 색인구조의 유지비용을 많이 필요로 한다. 이러한 색인구조의 유지비용은 색인을 유지하는 경로의 길이에 비례하기 때문에, 경로의 길이가 매우 길 때는 경로를 여러 개의 서브 경로로 나누어서, 서브 경로별로 여러 개의 색인구조를 유지함으로써 유지비용을 줄일 수 있다.

다음은 엘리먼트의 빠른 탐색을 위한 다차원 중포 엘리먼트 색인기법의 MNI와 MPI의 운용 알고리즘이다.

그림 2는 경로 EP에서 임의의 엘리먼트 E_i에서 중포

엘리먼트의 값이 E_{i+1}에서 E'_{i+1}로 변경될 경우의 MNI의 운용 알고리즘이다.

단계1	엘리먼트 E_{n-1}로부터 중포 엘리먼트까지 경로상의 타입 식별자 값들과 합쳐 키값으로 구성된 색인키 리스트들의 집합 S(A)를 구함(엘리먼트 E_{n-1}로부터 경로를 따라 순방향 운행)
단계2	엘리먼트 E'_{n-1}로부터 중포 엘리먼트까지 경로상의 타입 식별자 값들과 합쳐 키값으로 구성된 색인키 리스트들의 집합 S(B)를 구함. 만약, S(A) = S(B)이면 색인의 갱신이 필요 없으며, 그렇지 않으면 단계 3을 시행함
단계3	E_i를 직접 또는 간접적으로 참조하는 타겟타입 계층 T_i의 엘리먼트 Eid들의 집합 Q를 구함(엘리먼트 E_n로부터 경로를 따라 역방향 운행)
단계4	① S(A) ⊃ S(B) 이면, S(A) - S(B)를 β로 하고, β에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 Q에 있는 Eid들을 제거 ② S(A) ⊂ S(B) 이면, S(B) - S(A)를 β로 하고, β에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 Q에 있는 Eid들을 찾기 ③ 그렇지 않으면, S(A) - S(B)를 β1, S(B) - S(A)를 β2라 하고, β1에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 Q에 있는 Eid들을 제거하고, β2에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 Q에 있는 Eid들을 찾기

(그림 2) MNI의 운용 알고리즘

그림 3은 경로 EP에서 임의의 엘리먼트 E_i에서 중포 엘리먼트의 값이 E_{i+1}에서 E'_{i+1}로 변경될 경우의 MPI의 운용 알고리즘이다.

단계1	엘리먼트 E_{n-1}로부터 중포 엘리먼트까지 경로상의 타입 식별자 값들과 합쳐 키값으로 구성된 색인키 리스트들의 집합 S(A)를 구함(엘리먼트 E_{n-1}로부터 경로를 따라 순방향 운행)
단계2	엘리먼트 E'_{n-1}로부터 경로에 따른 순방향 운행을 통하여 중포 엘리먼트까지 서브경로 인스턴스들의 집합 SP(B)와 경로상의 타입 식별자 값들과 키값으로 구성된 색인키 리스트들의 집합 S(B)를 구함
단계3	S(A)에 있는 각 색인키 리스트에 해당하는 색인 레코드를 액세스하여 i번째 항목이 E_i이고 i+1번째 항목이 E_{i+1}인 경로 인스턴스를 삭제함과 동시에, 각 경로 인스턴스의 첫번째 항목에서 i번째 항목까지의 부분을 W에 보관함
단계4	i번째 항목이 E_i이고 i+1번째 항목이 E'_{i+1}인 새로운 경로 인스턴스 집합 P를 생성함. 이는 W에 있는 요소들과 SP(B)에 있는 요소들을 각각 연결함으로써 얻을 수 있음
단계5	S(B)에 있는 각 색인키 리스트에 해당하는 색인 레코드에 집합 P에 있는 경로 인스턴스를 추가함

(그림 3) MPI의 운용 알고리즘

4. 다차원 중포 엘리먼트 색인구조의 할당

제 4.1절에서는 하나의 경로에 대한 각 색인구조의 경로 길이에 따른 색인구조를 할당하는 기준을 제시한다. 그리고 제 4.2절에서는 두 개의 경로에 대한 여러 개의 술어들을 보다 일반적인 질의에 대해서 MPI를 사용한 질의처리 방법을 분석하여 최적의 색인할당방법을 제시한다.

4.1 단일 술어에 대한 할당

검색 질의에 대한 두 색인구조의 성능에 대해서는 MNI가 MPI에 비해 좋은 성능을 가진다. 이는 MPI에서는 색인 엔트리를 색인된 중포 엘리먼트의 경로 인스턴스(Eid의 리스트)들로 구성하기 때문에 색인 엔트리를 타켓 타입 계층에 속하는 엘리먼트 식별자만으로 구성하는 MNI에 비하여 많은 저장 공간의 오버헤드 때문이다. 그러나, 두 색인구조의 운용에 따른 유지비용에 대한 오버헤드는 색인된 중포속성의 경로 길이에 따라 많은 차이를 보이게 된다.

술어가 있는 경로의 길이가 2이고 역 참조자가 데이터베이스의 엘리먼트 내에서 제공될 경우에는 MNI와 MPI의 유지비용은 같게 된다. 그 이유로서, 먼저 경로상의 두 번째 타입 계층에 있는 엘리먼트 *E2*에서 색인된 애트리뷰트 *A2*가 변경되면 엘리먼트 E2내에 역 참조자가 존재하기 때문에 엘리먼트 *E2*를 참조하는 첫 번째 계층의 엘리먼트를 결정하기 위한 역방향 운행이 필요 없기 때문이다. 그리고, 술어가 있는 경로의 길이가 3이상인 경우에 MNI에서는 역 방향 운행이 필요하게 되므로 MPI에 비해 유지비용이 증가하게 된다. MNI의 유지비용을 지배하는 것은 역방향 운행에 의한 것으로, 역방향 운행에 필요한 엘리먼트의 액세스 개수는 엘리먼트 참조 공유도에 의해 결정된다. MPI에서는 역방향 운행이 필요 없기 때문에 MNI에서보다 유지비용이 매우 적게 된다.

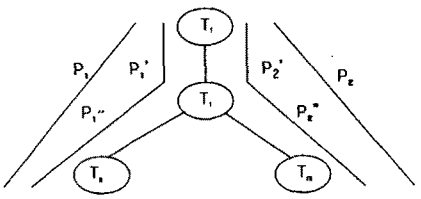
따라서, 위와 같은 분석의 결과로서 다음과 같은 결론을 얻을 수 있다. 첫째로 색인을 구축할 경로의 길이가 2인 경우에는 MNI가 적합하다. 이것은 유지비용은 두 가지 색인구조 모두 비슷한 반면에 MNI에서 검색 성능이 좋기 때문이다. 둘째로 색인을 구축할 경로의 길이가 3이상인 경우에는 일반적으로 MPI가 적합하다. 이것은 MPI의 검색 성능이 MNI에 필적하는 반면에 데이터베이스 변경에 의한 유지비용이 적게 되기 때문이다. 그리고 MPI는 각 엘리먼트 내에 역 참조자가 존재하지 않을 경우에도 사용이 가능하다.

4.2 두 개의 술어에 대한 할당

두 개의 술어로 된 그림 4는 집단화 계층에 따른 경로 구성을 본 논문에서 앞으로 사용할 경로들의 이름과 함께 나타낸 것이다. 그림 4는 분리된 경로에서 분리가 시작되는 타입 계층은 Ti이다. 그리고, 색인구조를 이용하지 않는 순방향 운행법(FT : Forward Traversal)에 의한 질의처리 비용식은 다음 식(1)과 같으며, 앞으로 각 색인할당 방법에서 질의처리 비용을 계산하기 위하여 이 식을 이용한다.

$$Cost\_FT = PC(T_i) + 2 \times N_i \times (n-i) \qquad (1)$$

여기서, $PC(T_i)$는 타입 T_i의 엘리먼트들을 가지는 디스크 페이지의 개수($1 \le i \le n$)이다.



(그림 4) 분리된 경로구성 스키마.

(1) 색인할당 방법(1): P_1과 P_2에 MPI색인을 할당하는 경우

이 색인할당 방법의 경우는 두 경로 P_1과 P_2 모두 분리하지 않고 색인을 할당하는 경우이다. 이 경우의 질의처리전략은 다음과 같다.

① $pred_n$을 평가하기 위해 경로 P_1상의 색인을 액세스한다.
② $pred_m$을 평가하기 위해 경로 P_2상의 색인을 액세스한다.
③ ①과 ②의 평가결과를 교집합한다.

색인할당 방법(1)의 비용식은 다음과 같이 나타낼 수 있다.

$$Cost(1) = MPA\_Cost(P_1) + MPA\_Cost(P_2) \qquad (2)$$

여기서, MPA_Cost는 MPI 색인구조의 액세스 비용이다.

(2) 색인할당 방법(2): P'_1, P''_1과 P''_2에 MPI색인을 할당하는 경우

이 색인할당 방법의 경우는 두 경로 모두 분리하여 색인을 할당하는 경우이다. 여기에서 두 경로의 첫 번째 부경로의 색인들은 동일하다. 이 경우의 질의처리 전략은 다음과 같다.

① $pred_n$을 평가하기 위해 경로 P''_1상의 색인을 액세스한다.
② $pred_m$을 평가하기 위해 경로 P''_2상의 색인을 액세스한다.
③ ①과 ②의 평가결과를 교집합한다.

④ 경로 P'_1상의 색인을 액세스 하여 최종결과를 얻는다.

색인할당 방법(2)경우의 질의처리전략들의 비용식은 다음과 같다.

$Cost(2) = MPA\_Cost(P''_1) + MPA\_Cost(P''_2) + no \times MPA\_Cost(P'_1)$ (3)

여기서, $no = \lceil k(i, n)/D'_m \rceil$ 로써 두 개의 술어 $pred_n$과 $pred_m$을 동시에 만족하는 엘리먼트의 수를 나타낸다.

(3) 색인할당 방법(3): P_1, P'_2와 P''_2에 MPI색인을 할당하는 경우

이 색인할당 방법의 경우는 경로 P_1, P'_2와 P''_2에 색인을 할당하는 것이다. 이 경우의 질의처리전략은 다음과 같다.

① $pred_n$을 평가하기 위하여 경로 P_1상의 색인을 액세스한다.

② $pred_m$을 평가하기 위하여 경로 P'_2상의 색인과 경로 P''_2상의 색인을 차례로 액세스한다.

③ ①과 ②의 평가 결과를 교집합한다.

색인할당 방법(3)경우의 질의처리전략의 비용식은 다음과 같다.

$Cost(3) = MPA\_Cost(P''_2) + no \times MPA\_Cost(P'_2) + MPA\_Cost(P_1)$ (4)

여기서, $no = k'(i, m)$로서 $pred_m$을 만족하는 타입 T_i의 엘리먼트 수이다.

(4) 색인할당 방법(4): P_1과 P''_2에 MPI색인을 할당하는 경우

이 색인할당 방법의 경우는 주어진 두 개의 겹침 경로 P_1과 P_2에서 경로 P_1은 분리하지 않고 색인을 할당하고, P_2는 분리하였지만 P''_2에만 색인을 할당할 경우이다. 이 경우의 질의처리전략은 다음과 같다.

① $pred_n$을 평가하기 위하여 경로 P_1상의 MPI색인을 액세스하여 (E_i, E_j)쌍들의 집합을 프로젝션하여 구한다.

② $pred_m$을 평가하기 위하여 경로 P''_2상의 색인을 액세스하여 E_i엘리먼트들의 집합을 구한다.

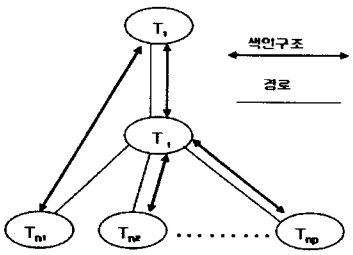
③ ①의 결과에서 두번째 항이 ②의 결과에 있는 첫번째 항을 구한다.

색인할당 방법(4)경우의 질의처리 전략의 비용식은 다음과 같다.

$Cost(4) = MPA\_Cost(P_1) + MPA\_Cost(P''_2)$ (5)

(5) 각 색인구조의 할당에 따른 질의처리 비용식의 비교

색인할당 방법(1), (2), (3), (4)에 대한 질의처리 비용식을 비교한 결과로 두 경로 모두에 색인을 할당하는 경우에는, 두 경로 모두 부경로로 분리하지 않는 색인할당 방법(1)이 두 경로 모두 또는 한 경로를 분리하는 색인할당 방법(2), (3)보다 질의처리 비용이 적다. 그리고 한 경로를 분리하여 공통의 부경로(P'_2)에는 색인을 할당하지 않는 색인할당 방법(4)가 두 경로 모두 부경로로 분리하지 않는 색인할당 방법(1)보다 질의처리 비용이 적다. 따라서 가장 최적의 색인할당방법은 색인할당방법(4)이다. 그림 5는 색인할당 방법(4)의 색인구성을 일반화한 것이다.



(그림 5) 색인할당 방법(4)의 일반화

5. 결론

본 논문에서는 XML Schema가 가지는 타입상속 개념을 고려한 XML질의들에 대한 처리를 효율적으로 지원할 수 있는 다차원 중포 색인구조와 다차원 경로 색인구조의 운용법을 제시하였다. 그리고 색인구조의 유지비용을 줄이기 위하여 저장 공간 및 갱신유지 비용을 최소화할 수 있는 효과적인 색인할당 방법을 제시하였다.

제시된 방법의 타당성을 확인하기 위하여 하나의 경로를 대상으로 각 색인구조의 운용에 따른 유지비용을 비교하였다. 비교 결과로 경로의 길이가 2인 경우는 다차원 중포색인을 할당하고, 경로의 길이가 3인 경우는 다차원 경로색인을 할당하는 것이 가장 효율적이었다. 특히 경로의 길이가 4이상일 경우에는 경로의 길이를 1, 2, 또는 3인 부경로들로 구성할 수 있으므로 각 부경로 별로 상기에서 제시한 색인구조를 할당한다.

한편 상하 타입 간에 두 개의 겹침 경로를 가지는 경우에 색인할당 방법을 분류해서, 각각에 대한 질의처리전략과 비용식을 제시하였다. 제시된 비용식을 이용하여 각 계층 간의 공통 부경로에는 색인을 할당하지 않는 것이 가장 효과적인 색인할당 방법임을 알 수 있었다. 이는 질의 처리 시 색인 엔트리에 프로젝션 연산을 이용함으로써 비용을 최소화 할 수 있기 때문이다. 또한 상기 결과를 기반으로 하여 세 개 이상의 경로를 대상으로 하는 일반적인 색인할당 방법을 제시하였다. 즉, 여러 경로가 겹치는 경우에는 엘리먼트 참조 공유도가 가장 낮은 경로에 대해서 다차원 경로 색인을 할당한다. 그리고 나머지 경로에는 두 개의 부경로로 분리하여 비 겹침 경로에만 색인을 할당하는 것이 가장 효율적으로 질의처리를 지원한다는 것을 알 수 있었다.

참고문헌

[1] Extensible Markup Language(XML) 1.0, *http://www.w3.org/TR/1998/-REC-xml-19980210.*

[2] XML Schema Part 0:Primer, *http://www.w3.org/TR/xmlschema-0,* Oct. 2004.

[3] Arnaud Le Hors, et al., Document Object Model Level2 Core. W3C Recommendation, 2000.

[4] 이성원, 이종학, "다차원 파일구조를 이용한 XML 데이터베이스의 중포 엘리먼트 색인기법," *한국멀티미디어학회 2005 추계 학술발표대회,* pp. 10-12. 2005년 4월.

[5] XQuery: An XML Query Language, *http://www.w3.org/TR/xquery,* Sep. 2005.

[6] XPath: XML Path Language, *http://www.w3.org/TR/xpath,* Nov. 1999.

[7] Goldman, R. and Widom, J. "DataGuides: Enable Query Formulation and Optimization in Semistructured DataBases," In *Proceedings of the International Conference on Very Large Data Bases,* Athens, Greece, pp. 436-445, Aug. 1997.

[8] Milo, T. and Suciu, D. "Index Structures for Path Expression," In *Proceedings of the International Conference on Database Theory,* pp.277-295, Jan. 1999.

[9] Cooper, B. et al., "A Fast Index for Semistructured Data," In *Proceedings of the International Conference on Very Large Data Bases,* Rome, Italy, pp. 341-350, Sep. 2001.

[10] Chung, C. W. and Min, J. K., "APEX: An Adaptive Path Index for XML Data," In *Proceedings of the ACM SIGMOD International Conference on the Management of Data,* 121-132, 2002.

[11] Whang, K. Y. and Krishnamurthy, R., "The Multilevel Grid File - A Dynamic Hierarchical Multidimensional File Structure," In *Proceedings of the International Conference on Database Systems for Advanced Applications(DASFAA),* Tokyo, pp. 449-459, Apr. 1991.