

내포 질의의 효율적 평가를 위한 분리 색인 기법

(A Separated Indexing Technique for Efficient Evaluation of Nested Queries)

權 用 武*, 朴 容 震*

(Yong Moo Kwon and Yong Jin Park)

要 約

본 논문에서는 객체 지향 모형에서 집약 계층에 대한 내포 질의의 효율적 평가를 위한 새로운 색인 기법을 제안한다. 색인 자료구조로서 검색되어지는 인스턴스 식별자 및 갱신을 위한 경로 정보를 각각 리프 노드 및 부리프 노드에 분리하여 관리하는 확장된 B⁺ 트리를 제시한다. 또한 제시된 색인 자료 구조에서의 검색 및 갱신 알고리즘을 제시한다. 검색 및 갱신을 위한 전체비용면에서 기존 색인 기법과 성능 비교를 통해 제안된 색인 기법의 우수성을 보인다.

Abstract

In this paper, a new indexing technique is proposed for efficient evaluation of nested queries on aggregation hierarchy in object-oriented data model. As an index data structure, an extended B⁺ tree is introduced in which instance identifier to be searched and path information used for update of index record are stored in leaf node and subleaf node, respectively. the retrieval and update algorithm on the introduced index data structure is provided. Comparisons under a variety of conditions are given with current indexing techniques, showing improved performance in cost, i.e., the total number of pages accessed for retrieval and update.

I. 서 론

다중 매체 정보는 객체 지향 개념에 근거하여 집약 계층(aggregation hierarchy) 및 계승 계층(inheritance hierarchy)으로 모형화되며^[1, 2] 이들 계층에 대한 질의^[3]의 빠른 평가를 위해서는 효율적인 색인 기법이 필수적으로 요구된다.

다중 매체 정보간의 is-part-of 관계를 나타내는 집약 계층에 대한 질의는 내포된 속성을 이용한 내포 질의^[3] 특성을 가진다. 내포 질의를 위한 색인 기법은 색인 키로

서 역시 내포된 속성 값을 이용해야 한다는 개념이 제시되었으며^[4] 그 후 색인의 검색 및 갱신비용을 고려하여 내포 색인(nested index) 기법, 경로 색인(path index) 기법 및 다중 색인(multi-index) 기법이 제안되고 각 색인 기법의 성능이 평가 비교되었다^[5]. 참고문헌 [5]의 성능 평가 결과에 의하면 내포 색인 기법은 검색비용면에서 우수하나 큰 갱신비용을 가지는 단점을 가지며 경로 색인 기법은 갱신비용은 적으나 큰 검색비용을 가지는 단점을 가진다. 다중 색인 기법은 가장 작은 갱신비용을 가지나 여러 색인들을 룩업(lookup)해야함에 의해 가장 큰 검색비용을 가지는 단점을 가진다. 그 결과 참고문헌 [5]에 의하면 이들 색인 기법은 경로 길이 2 및 3의 경로에만 적합하며 경로 길이 4이상의 경우에는 경로를 경로 길이 2 및 3의 부경로로 분할한 후 각 부경로에 적합한 색인을 할당하는 방법이 권고되고 있다. 이 권고된 방법은 갱신을

* 正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)
接受日字: 1991年 12月 12日

부경로내로 한정성킴에 의해 갱신비용을 감소시키나 다중 색인 기법과 같이 여러 색인들을 록업해야함에 의해 역시 큰 검색비용을 가지는 단점을 가진다.

다중 매체 정보간의 is-a 관계를 나타내는 계승 계층(inheritance hierarchy)에 대한 질의^[3]의 효율적 평가를 위한 색인 기법으로는 클래스 계층 색인 기법^[6]이 제안되어 있다.

집약 계층 및 계승 계층 모두에 대한 질의를 지원하기 위한 연구로는 일반화된 색인(generalized index) 기법^[7] 및 내포-계승(nested-inherited) 색인 기법^[8]이 제안되어 있다. 이들 색인 기법의 기본 개념은 집약 계층에 대한 기존 색인 기법 및 계승 계층에 대한 클래스 색인 기법을 결합한 것이다. 일반화된 색인 기법은 내포 색인 기법과 클래스 계층 색인 기법을 결합한 기법으로서 집약 계층을 위해 내포 색인 기법을 사용함에 의해 역시 갱신 비용에 문제점이 존재한다. 내포-계승 색인 기법은 집약 계층을 위해 기본색인(내포 색인에 해당됨) 및 보조 색인(다중 색인에 해당됨)을 결합하여 사용하며 또한 계승 계층을 위해 기본 색인 및 보조 색인 각각에 클래스 계층 색인을 결합하였다. 내포-계승 색인 기법은 갱신시 보조 색인을 이용함에 의해 내포 색인 기법의 단점인 후방 탐색비용^[5]을 감소시켰으나 갱신시 여러개의 보조 색인에 대한 록업 동작이 요구됨에 의해 역시 갱신비용에 문제점이 존재한다.

다중 매체 정보는 자료 모형 방법에 따라 집약 계층 및 계승 계층 모두 존재하게 되며 이 경우 색인 구성 방법은 집약 계층에 대한 색인 기법 및 계승 계층에 대한 색인 기법의 결합된 형태에 의해 실현되어진다. 본 논문에서는 다중 매체 정보에 대한 질의를 위한 색인 기법중 집약 계층에 대한 색인 기법에 대해 연구한다. 즉 집약 계층에 대해 제안된 기존 색인 기법의 단점을 개선하여 검색 및 갱신을 위한 전체비용을 감소시킨 분리 색인(separated index) 기법을 제안한다. 또한 제안된 색인 구성 방법의 특성을 바탕으로 개선된 갱신 알고리즘을 제안한다.

II장에서는 내포 질의의 개념에 대해 기술하고 III장에서는 기존 색인 기법의 문제점을 고찰한다. IV장에서는 분리 색인 기법을 제안한다. V장에서는 제안된 색인 기법의 비용모형을 제시하고 VI장에서는 기존 색인 기법과의 성능 비교를 수행한다.

II. 기본 개념

객체 지향 자료 모형에서 is-part-of 관계를 나타내는 집약 계층상에 존재하는 하나의 경로 P는 다음과 같이 정의된다.

[정의] 경로 P

경로 P는 $P = C(1) A(1) A(2) \dots A(i) \dots A(n)$ 로 정의된다.

경로 P에서 C(1)은 루트 클래스이다. A(i)는 클래스 C(i)의 속성이다($1 \leq i \leq n$). 클래스 C(i)는 클래스 C(i-1)의 속성 A(i-1)의 도메인이다($1 < i \leq n$). 클래스 C(n)은 경로 P의 리프 클래스이다. 경로 P의 경로 길이 pl은 경로내에 존재하는 클래스 수로서 리프 클래스가 C(n)인 경우 n이 된다.

경로 P에서 리프 클래스 C(n)의 속성 A(n)은 리프 클래스에 대해서는 직접 포함된 단순 속성(simple attribute)이지만 리프 클래스를 제외한 다른 모든 클래스에 대해서는 간접적으로 포함된 내포 속성(nested attribute)이 된다. 이와 같이 내포 속성을 이용한 질의를 내포 질의라 하고 단순 속성을 이용한 질의를 단순 질의라 한다.

경로 P의 실제 예는 $I(1) I(2) \dots I(i) \dots I(n) V(n)$ 으로 표시된다. 여기서 I(i)는 클래스 C(i)에 속하는 인스턴스이다. 또한 인스턴스 I(i)는 인스턴스 I(i-1)의 속성 A(i-1)의 실제 값이며($1 < i \leq n$) V(n)은 인스턴스 I(n)의 속성 A(n)의 실제 값이다.

색인 레코드 구축시 인스턴스 및 경로 정보는 해당 인스턴스의 IID(Instance Identifier)를 이용하여 표시한다. 본 논문에서는 인스턴스 자체와 인스턴스에 대한 IID를 구별하기 위해 인스턴스 I(i)의 IID를 U(i)로 표시한다. 특히 IID들 중 루트 클래스 인스턴스 I(i)에 대한 IID인 U(1)을 RID(Root class instance Identifier)라 하고 리프 클래스 인스턴스 I(n)에 대한 IID인 U(n)을 LID(Leaf class instance Identifier)라 한다. 또한 경로 $I(i) I(i-1) \dots I(j)$ 에 대한 IID 리스트인 $U(i) U(i-1) \dots U(j)$ 를 P(i, j)라 표시한다. P(i, j)에서 i는 시작 IID의 해당 클래스가 C(i)이고 j는 마지막 IID의 해당 클래스가 C(j)임을 나타낸다.

III. 문제점 고찰

집약 계층에서 색인 구축 대상이 되는 경로가 선택되는 경우 검색 대상 클래스는 선택된 색인 키에 대해서 루트 클래스가 된다. 한편 경로에 존재하는 임의의 인스턴스의 속성값이 바뀌는 경우 색인 레코드의 갱신이 요구되며 이를 위해서는 색인 키로부터 루트 클래스까지의 경로 정보가 요구된다. 예를들면 경로 정보 $U(1) U(2) \dots U(i) \dots U(n) V(n)$ 에 대해 색인을 구성하는 경우 속성 값 V(n)은 색인 키 값이 되며 검색 대상 정보는 U(1)이 된다. 또한 색인 레코드 갱신을 위한 정보는 색인 키 값 V(n)과

U(1)의 연결 정보인 U(n) U(n-1).. U(i)..U(2)이다. 즉 색인 구성시 요구되는 정보는 검색 대상에 대한 정보인 RID와 갱신을 위해 필요한 경로 정보 P(n, 2)로 분류할 수 있다.

집약 계층을 위한 기존 색인 기법의 색인 구성 방법을 검색 대상 및 갱신을 위한 경로 정보 관점에서 고찰해보면 다음과 같다.

내포 색인 기법은 색인 레코드에 검색 대상 정보인 RID들만을 포함한다. 따라서 기존 색인 기법 중 가장 작은 검색비용을 가진다. 그러나 갱신을 위한 정보는 색인 레코드에 포함되어 있지 않다. 그러므로 색인 레코드 갱신을 지원하기 위해 각각의 인스턴스 내에 후방참조정보^[5]를 가져야하며 색인 레코드 갱신시 경로 정보를 구하기 위해 각각의 인스턴스들을 접근해 후방탐색을 수행해야 하므로 매우 큰 갱신비용이 요구되는 단점이 있다.

경로 색인 기법은 경로 정보를 레코드내에 포함함에 의해 후방탐색의 필요성이 없어졌으나 검색 대상 정보인 RID와 갱신을 위한 경로 정보인 P(n, 2)를 구분하지 않고 전체 경로에 대한 정보 P(n, 1)를 동일 색인 레코드에 저장한다. 즉 U(n) U(n-1)..U(i)..U(2) U(1)의 형태의 경로 정보를 레코드에 포함한다. 따라서 검색시 접근 단위 역시 P(n, 1) 단위로 이루어짐에 의해 경로 길이가 길어지는 경우 이에 비례하여 검색비용이 커지는 단점을 가진다.

다중 색인 기법은 각 클래스별로 단순 속성 값을 사용하여 색인을 구성하는 기법으로서 갱신은 갱신이 일어난 클래스의 색인에서만 수행되므로 효율적이나 검색시 여러개의 색인을 룩업해야함에 의해 검색비용이 매우 큰 단점을 가진다.

내포-계승 색인 기법은 검색 대상 정보 및 갱신을 위한 정보를 모두 포함하고 있으며 색인 구성의 기본 개념은 내포 색인에 해당되는 기본 색인(primary index) 및 다중 색인에 해당되는 보조 색인(auxiliary index)을 결합한 것이다. 이 기법은 후방탐색시 보조 색인을 사용함에 의해 내포 색인 기법에 비해 갱신비용을 감소시켰다는 장점이 있다. 그러나 이 기법은 갱신 대상 정보를 구하기 위해 여러개의 보조 색인들을 룩업해야한다. 룩업동작에 요구되는 비용은 내포 색인 기법에서와 같이 후방탐색시 각각의 인스턴스에 접근하는 비용에 비해서는 작은 비용을 가지나 다중 색인 기법이 큰 검색비용을 가지는 것과 같이 역시 갱신비용에 문제점이 존재한다. 또한 기본 색인 레코드는 내포 색인 레코드와 동일한 특성을 가짐으로 인해 VI장에서 지적하는 바와 같이 갱신 대상 정보가 색인 레코드내에서 무작위로 존재하는 단점을 가진다.

본 논문에서는 이와 같은 문제점을 고려하여 검색 대상 정보 및 갱신을 위한 정보를 분리하여 관리하고 갱신을

위한 정보로는 색인 룩업 없이 바로 이용 가능한 경로 정보 P(n, 2)를 가지며 또한 색인 레코드에서 갱신 대상 정보가 연속적으로 존재하는 색인기법을 IV장에서 제안한다.

IV. 분리 색인 기법

분리 색인 기법의 기본 구조는 그림 1에 나타난 바와 같이 기존 B⁺ 트리^[6]를 확장한 구조를 가지며 다음과 같은 특성을 가진다.

- (i) 비리프 노드, 리프 노드와 부리프 노드로 구성된다. 단 부리프 노드는 경로 길이가 2이상인 경우에만 존재한다.
- (ii) 비리프 노드 및 리프 노드는 기존 B⁺트리 구조 및 특성을 가진다.
- (iii) 리프 노드는 색인 키, 부리프 노드 페이지포인터 및 색인 키를 내포 속성으로 갖는 검색대상 정보를 가진다.
- (iv) 부리프 노드는 갱신을 위한 정보로서 리프 노드의 검색 대상 정보 각각에 대한 경로정보를 가진다.

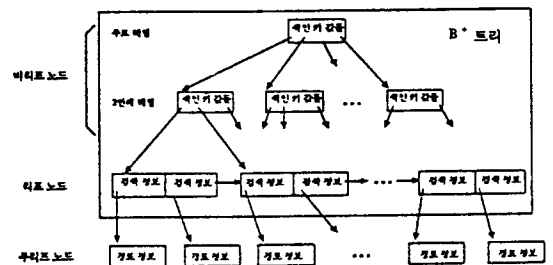


그림 1. 분리 색인 기법 기본구조
Fig. 1. Basic structure of separated indexing technique.

1. 색인 구성

분리 색인 기법의 자료구조는 그림 2와 같다.

(1) 비리프 노드

비리프 노드 자료 구조는 기존 B⁺ 트리와 동일하다.

(2) 리프 노드 레코드

리프 노드 레코드의 자료 구조는 검색 대상인 RID 정보를 포함하며 갱신을 위한 경로 정보를 갖고 있는 부리프 노드 레코드의 페이지 포인터를 갖는다. 범람 정보에서 SNO(Sequence Number of Ordering)는 RID 순차번호로서 해당 레코드에서 RID들을 정렬(ordering) 순서에 의해 정렬한 후 1부터 순차적으로 각 RID에 부여되는

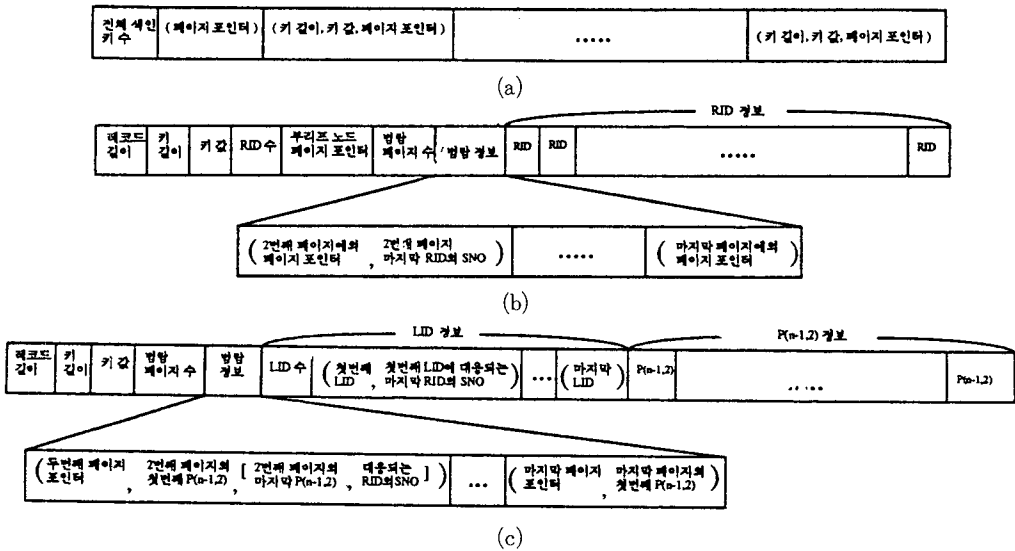


그림 2. 분리색인기법 자료 구조

(a) 비리프 노드 (b) 리프 노드 레코드 (c) 부리프 노드 레코드

Fig. 2. Data structure of separated indexing technique.

(a) nonleaf node, (b) leaf node record, (c) subleaf node record.

번호이다. 따라서 레코드에서 첫번째 존재하는 RID의 SNO는 1이며 가장 마지막에 존재하는 RID의 SNO는 전체 RID 수가 된다. RID 정렬 순서는 (3) 부리프 노드 레코드에서 LID 및 경로 정보 P(n-1, 2)의 정렬 순서와 함께 기술한다.

(3) 부리프 노드 레코드

부리프 노드 레코드는 리프 노드 레코드에 존재하는 RID에 대한 경로 정보 P(n, 2)를 가진다. 경로 정보 P(n, 2)들은 동일한 LID를 갖는 경로 정보 집합 {P(n-1, 2)} 단위로 묶어 저장한다. 즉 그림 2(c)에 나타낸 바와 같이 LID 정보와 P(n-1, 2) 경로 정보를 분리하고 우선 LID 정보를 저장한 후 그 다음에 P(n-1, 2) 경로 정보를 저장한다. 부리프 노드 레코드에서 RID에 대한 SNO는 LID, P(n-1, 2), RID의 관계 정보로서 이용된다.

LID, 경로 정보 P(n-1, 2) 및 RID 정렬 순서는 다음과 같다.

우선 LID 정보에서 {LID, LID를 속성 값의 식별자로 갖는 마지막 RID의 SNO}들을 해당 LID의 올림차순으로 정렬한다.

그 다음 경로 정보 P(n-1, 2)를 저장하며 P(n-1, 2)들의 정렬 순서는 대응되는 LID의 정렬 순서에 의해 정해진다. 즉 동일 LID에 대응되는 P(n-1, 2) 집합 단위로 LID 정렬 순서와 동일하게 순차적으로 저장한다. 또한 동일 LID를 갖는 경로 정보 P(n-1, 2) 집합내에 존재하는 경로 정보 P(n-1, 2)들의 정렬 순서는 다음과 같다. 우선

P(n-1, 2)에 존재하는 U(n-1)의 올림차순, U(n-1)이 같은 경우는 U(n-2)의 올림차순, 최종적으로는 U(2)의 올림차순으로 정렬한다.

리프 노드 레코드에서 RID 정렬 순서는 부리프 노드의 대응되는 경로 정보 P(n-1, 2)의 정렬 순서와 동일하다. 즉 각 경로 정보 P(n-1, 2)는 리프 노드 레코드의 각 RID에 1:1로 대응되며 따라서 RID는 대응되는 경로 정보 P(n-1, 2)의 정렬 순서와 동일하게 정렬한다. 단 동일한 P(n-1, 2)에 대응되는 RID가 여러개 존재하는 경우는 이들 RID 집합 내에서 RID의 올림차순으로 정렬한다.

2. 검색 및 갱신 알고리즘

(1) 검색 알고리즘

단계 1: 비리프 노드를 접근하여 색인 키에 대한 리프 노드 레코드의 페이지 주소를 구한다.

단계 2: 리프 노드 레코드를 접근하여 검색 대상 정보인 RID를 구한다. 만약 레코드 크기가 한 페이지를 넘는 경우 범람 정보를 이용하여 연속적으로 존재하는 페이지 접근을 통해 검색 대상 정보인 RID들을 구한다.

(2) 갱신 알고리즘

색인이 구성된 경로에서 인스턴스 I(i)의 속성 값이 처음 I(i+1)에서 I'(i+1)로 바뀌는 경우 갱신은 다음과 같이 수행된다.

단계 1: 인스턴스 I(i)의 갱신

인스턴스 $I(i)$ 를 접근하여 인스턴스 $I(i)$ 의 속성 값을 $I(i+1)$ 에서 $I'(i+1)$ 로 갱신한다.

단계 2: 전방탐색

$I(i)$ 의 처음 속성 값 $I(i+1)$ 로부터 $I(n)$ 까지의 경로 정보 및 $I(n)$ 의 속성 값 $V(n)$ 을 구하고, 바뀐 속성 값 $I'(i+1)$ 로부터 $I'(n)$ 까지의 경로 정보 및 속 $I'(n)$ 의 속성 값 $V'(n)$ 을 구한다. 전방탐색은 다음과 같이 수행한다. $I(i+1)$ 및 $I'(i+1)$ 을 접근하여 각각에 대한 속성 값 $I(i+2)$ 와 $I'(i+2)$ 를 구한후 두 속성 값을 비교한다. 서로 일치하지 않는 경우는 전방탐색을 동일한 방법으로 계속한다. 만약 전방탐색 도중에 두 속성 값이 동일한 경우는 하나의 속성 값에 대해서만 전방탐색을 수행한다.

구해진 2개의 경로에서 $P.old=U(i)U(i+1)U(i+2) \dots U(n)$, $P.new=U(i)U'(i+1)U'(i+2) \dots U'(n)$ 라고 한다.

단계 3: 색인 레코드 갱신

색인 레코드 갱신은 기본적으로 삭제와 삽입에 의해 수행된다. 즉 $U(i+1)$ 에 대한 색인 키 값 $V(n)$ 레코드에서 삭제 동작 및 $U'(i+1)$ 에 대한 색인 키 값 $V'(n)$ 레코드에서 삽입동작이 요구된다. 삭제 및 삽입 기본 절차는 우선 부리프 노드 레코드에서 LID 및 $P(n-1, 2)$ 정보를 갱신한 후 대응되는 RID의 SNO를 이용해 리프 노드 레코드에서 RID 정보를 갱신한다.

단계 3.1: 삭제 절차

(i) $V(n)$ 에 대한 리프 노드 레코드를 접근하여 부리프 노드 페이지 주소를 구한다.

(ii) 부리프 노드 레코드에서 $P.old$ 의 $U(n-1)U(n-2) \dots U(i)$ 를 포함하는 $P(n-1, 2)$ 들을 삭제한다. 삭제되는 각각의 $P(n-1, 2)$ 에서 $P(i-1, 2)$ 를 삽입 절차를 위해 메모리 상에 기억한다. $P.old$ 에 포함된 $U(n)$ 은 대응되는 $P(n-1, 2)$ 들이 모두 삭제되는 경우에 한해서만 삭제한다.

(iii) 단계 (ii)에서 삭제된 $P(n-1, 2)$ 에 대응되는 RID의 SNO를 이용하여 리프 노드 레코드에서 해당 RID들을 삭제한다. 이때 삽입 절차를 위해 삭제되는 RID들을 메모리 상에 기억한다.

단계 (ii)에서 삭제되는 $P(n-1, 2)$ 들이 언제나 연속적으로 존재한다는 특성을 이용하여 삭제 대상 $P(n-1, 2)$ 들을 포함한 페이지를 다음과 같이 분류하여 갱신한다. 즉 첫번째 삭제할 $P(n-1, 2)$ 를 포함하며 삭제 대상이 아닌 $P(n-1, 2)$ 도 포함된 페이지(FP: First part Page), 페이지내에 존재하는 모든 $P(n-1, 2)$ 가 삭제 대상인 페이지(WP: Whole Page), 마지막 삭제 대상 $P(n-1, 2)$ 를 포함하며 삭제 대상이 아닌 $P(n-1, 2)$ 도 포함한 페이

지(LP: Last part Page)로 분류한다. 그 후에 WP의 경우는 삭제시 레코드 디렉토리에서 범람정보만을 갱신함에 의해 삭제를 수행한다.

단계 (iii)에서 RID들을 삭제할때도 동일한 개념이 적용된다.

단계 3.2: 삽입 절차

(i) $V'(n)$ 에 대한 리프 노드 레코드를 접근하여 부리프 노드 페이지 주소를 구한다.

(ii) $P.new$ 및 $P(i-1, 2)$ 들을 이용하여 갱신된 경로 정보 $P'(n-1, 2)$ 를 구한다. 즉 $P'(n-1, 2)=U'(n-1) \dots U'(i+1)U(i)U(i-1) \dots U(2)$ 로 구성된다. 구해진 $P'(n-1, 2)$ 들 및 $P.new$ 에 존재하는 $U'(n)$ 을 부리프 노드 레코드에 삽입한다.

(iii) $V(n)$ 리프 노드 레코드에서 삭제된 RID들을 $V'(n)$ 리프 노드 레코드에 삽입한다.

단계 (ii)에서 새로운 경로 정보 $P'(n-1, 2)$ 들을 $V'(n)$ 레코드에 삽입 절차시 갱신되는 클래스가 $C(n-1)$ 인 경우 및 $C(n)$ 인 경우 WP에 존재하는 $P(n-1, 2)$ 들은 영향을 받지 않는다. 따라서 이 두 경우에는 삭제 절차에서와 같이 WP에 대해 레코드 디렉토리의 갱신만으로 처리한다. 그 이외의 경우는 WP를 접근하여 각각의 $P(n-1, 2)$ 를 $P'(n-1, 2)$ 로 갱신한다.

단계 (iii)에서 $V'(n)$ 의 리프 노드에 RID들을 삽입하는 경우는 WP내에 RID 만이 존재하므로 갱신되는 클래스의 위치에 관계없이 WP에 대해서는 언제나 레코드 디렉토리의 갱신만으로 삽입 절차를 수행한다.

한편 전방탐색에 의해 구해진 두개의 색인 키 값이 동일한 경우는 하나의 색인 키 값에 대한 레코드내에서 삭제 및 삽입 절차를 수행한다. 즉 부리프 노드 레코드의 경우 $P.old$ 에 대응되는 $P(n-1, 2)$ 들을 $P.new$ 에 대응되는 $P'(n-1, 2)$ 로 갱신한후 자료 정렬 순서를 고려하여 $P(n-1, 2)$ 들을 재구성한다. 리프 노드 레코드의 경우도 동일한 개념으로 갱신을 수행한다.

V. 비용 모델

표 1은 비용 모델에서 사용하는 변수 및 변수 길이를 나타낸다. 비용 모델을 구하기에 앞서 주어진 경로에 대해 다음과 같은 가정을 한다.

가정 (1) 클래스 $C(i)$ 의 인스턴스 수 $N(i)$ 는 클래스 $C(i)$ 의 속성 값의 수 $D(i)$ 에 대해 참고 공유도 $K(i)$ 값으로 균등하게 분포된다. 즉 $K(i)=N(i)/D(i)$ 가 성립한다. ($1 \leq i \leq n$).

가정 (2) 경로에 존재하는 모든 클래스에 대해 $D(i)=N(i+1)$ 이 성립한다. ($1 \leq i < n$).

가정 (3) 모든 색인 키 길이는 동일하다.

가정 (4) 모든 속성은 하나의 값만을 갖는다.

각각의 인스턴스 자료구조에서, 클래스 속성의 속성 값은 IID로 나타내지고 실제 인스턴스가 존재하는 페이지 주소는 참고문헌 [5]와 같이 IID와 해쉬(hash) 테이블을 사용하여 구한다. 따라서 각각의 인스턴스를 접근하기 위해서는 두번의 페이지 접근비용이 요구된다. 즉 한번은 해쉬 테이블 접근비용이고 또 한번은 해쉬 테이블에서 구한 인스턴스 위치 정보에 의해 실제 인스턴스를 접근하는 비용이다.

표 1. 변수 및 변수 길이
Table 1. Variable and Variable length.

변수	변수 의미	변수길이 (바이트)
rl	레코드 길이 필드의 길이	2
kll	색인 키 길이 필드의 길이	2
kl	색인 키 값 필드의 평균 길이	8
pp	페이지 포인터 필드의 길이	4
op	범람 페이지 수 필드의 길이	2
nRID	RID수 필드 길이	2
nLID	LID수 필드 길이	2
SN	RID순차 번호(SNO) 필드 길이	2
P	페이지 크기	4096
IIDL	인스턴스 식별자 길이	8

1. 저장비용

저장비용은 확장된 B+ 트리의 버리프 노드 저장비용, 리프 노드 저장비용 및 버리프 노드 저장비용의 합으로 주어진다. 버리프 노드의 평균 출력 분기수는 참고문헌 [5]에서와 동일하게 218을 사용한다. 버리프 노드 저장비용 모델은 참고문헌 [5]와 동일한 개념에 의해 구해지므로 본 논문에는 리프 노드 및 버리프 노드의 저장비용 모델만을 포함시킨다.

리프 노드 레코드 길이 XSL 및 버리프 노드의 레코드 크기 XSS는 각각 식 (1) 및 식 (2)로 주어진다. 식에서 FI는 기본적으로 포함되는 필드의 길이, OI는 범람 정보 길이, LI는 LID 정보 길이를 나타낸다. k(1, n)은 참고공유도 K(1)으로부터 K(n)까지의 적을 나타낸다.

$$XSL=FI+OI+k(1, n) \times IIDL \tag{1}$$

$$FI=rl+kl+kll+nRID+pp+op$$

$$OI=(p-1) \times (pp+SN) - SN$$

$$p=\lceil XSL/P \rceil$$

$$k(1, n) = \prod_{j=1}^n K(j)$$

$$XSS=FI+OI+LI+k(1, n) \times IIDL \times (n-2) \tag{2}$$

$$FI=rl+kl+kll+op$$

$$LI=nLID+K(n) \times (IIDL+SN) - SN$$

$$OI=(p-1) \times [pp+2 \times IIDL \times (n-2) + SN] - IIDL \times (n-2) - SN$$

$$p=\lceil XSS/P \rceil$$

리프 노드 또는 버리프 노드 레코드 길이를 대표하여 XS라 표시할 때 리프 노드 또는 버리프 노드의 저장비용 S는 식 (3)과 같이 구해진다. 식 (3)에서 D(n)은 클래스 C(n)의 서로 다른 속성 값의 수, 즉 서로 다른 색인 키 값의 수이다.

$$S = \begin{cases} \lceil D(n)/[P/XS] \rceil, & XS \leq P \text{인 경우} \\ D(n) \times \lceil XS/P \rceil, & XS > P \text{인 경우} \end{cases} \tag{3}$$

2. 검색비용

본 논문에서는 하나의 색인 키를 이용한 단일 질의만을 고려한다. 검색비용 RS는 식 (4)와 같이 버리프 노드 높이 h와 리프 노드 레코드 길이 XSL에 대한 페이지 수의 합으로 주어진다.

$$RS=h+\lceil XSL/P \rceil \tag{4}$$

3. 갱신비용

갱신비용은 인스턴스 I(i)의 속성 값이 I(i+1)에서 I'(i+1)로 갱신되는 경우 요구되는 비용으로서 전방탐색비용(CFT) 및 색인 레코드 갱신비용(CRU)의 합으로 주어진다. 인스턴스 I(i)의 속성 값 갱신비용은 인스턴스 자체의 갱신이므로 색인 갱신비용에는 포함시키지 않는다. 갱신비용 모델에서 i 및 n은 각각 갱신되는 클래스의 위치 및 리프클래스의 위치를 나타낸다.

(1) 전방탐색비용(CFT)

CFT는 식 (5)와 같이 구해진다. 인스턴스 I(i)의 갱신되기 이전 속성 값이 I(m)이고 갱신 후의 새로운 속성 값을 I'(m)이라 할 때 PAdiff(m)은 I(m)의 단순 속성 값이 I'(m)의 단순 속성 값과 다를 확률이다.

$$CFT = \begin{cases} 0, & i=n \text{인 경우} \\ 2 \times CFT(i+1, n), & i \neq n \text{인 경우} \end{cases} \tag{5}$$

$$CFT(i+1, n) = \sum_{m=i}^{n-1} [1 + PAdiff(m)]$$

$$PAdiff(m) = \begin{cases} 1, & m=i \text{인 경우} \\ 1 - \frac{[\prod_{j=i+1}^m K(j) - 1]}{[D(i) - 1]}, & i+1 \leq m \leq n \text{인 경우} \end{cases}$$

(2) 색인 레코드 갱신비용 (CRU)

CRU는 버리프 노드 접근비용(CBN), 부리프 노드 레코드 갱신비용(CBS)와 리프 노드 레코드 갱신비용 (CBL)의 합으로 주어진다.

(i) 버리프 노드 접근비용 (CBN)

CBN은 색인 키 값 $V(n)$ 의 리프 노드 레코드의 주소를 구하기 위한 버리프 노드 접근비용 h 와 $V(n) \neq V'(n)$ 인 경우 색인 키 값 $V'(n)$ 의 리프 노드 레코드의 주소를 구하기 위한 버리프 노드 접근비용 합으로 구해진다.

$$CBN = h \times [1 + PAdiff(n)] \quad (6)$$

(ii) 부리프 노드 갱신비용 (CBS)

CBS는 경로 정보 $P(n-1, 2)$ 의 삭제에 위한 비용 (NPDS)과 삽입을 위한 비용(NPIS)의 합으로 주어진다.

$$CBS = NPDS + NPIS \quad (7)$$

NPDS는 식 (8)로 주어진다. 첫번째 페이지는 삭제 대상 $P(n-1, 2)$ 가 없는 경우도 레코드 디렉토리 정보를 구하기 위해 항상 접근해야하므로 이에 대한 비용 $(1-Pf)$ 가 요구된다. Pf 는 삭제할 $P(n-1, 2)$ 가 첫번째 페이지에 존재할 확률이다. npd 값은 삭제할 $P(n-1, 2)$ 를 포함한 페이지의 접근비용 및 삭제후에 다시 저장하기위한 비용을 합한 것으로서 삭제대상 $P(n-1, 2)$ 를 포함한 페이지 수의 두 배로 주어진다. s 는 부리프 노드 레코드에서 삭제해야될 $P(n-1, 2)$ 의 수를 의미한다.

$$NPDS = npd + 1 - Pf \quad (8)$$

$$npd = \begin{cases} 2, & XSS \leq P \text{ 또는 } s=1 \text{인 경우} \\ 4, & \text{그 이외의 경우} \end{cases}$$

$$Pf = \begin{cases} 1/p, & GLS \leq P \text{인 경우} \\ 1/k(i, n), & GLS > P \text{인 경우} \end{cases}$$

$$GLS = s \times IIDL \times (n-2)$$

$$s = k(1, i-1) = \prod_{j=1}^{i-1} K(j)$$

$$P = \lceil XSS/P \rceil$$

NPIS는 삭제된 $P(n-1, 2)$ 들을 $P'(n-1, 2)$ 들로 갱신하여 새로운 색인 키 값 $V'(n)$ 에 대한 레코드에 삽입하는 비용이다. 이때 분리 색인 기법의 자료구조 특성으로 인해 삽입 위치는 한 곳으로 정해진다. 즉 삽입 위치를 기준

으로 일련의 $P'(n-1, 2)$ 들이 삽입되며 이때 삽입 정보의 크기에 의해 페이지 분리 및 재구성이 요구된다. 그러나 이에 대한 정확한 비용 모델은 제안된 분리 색인 기법의 특성 평가에 크게 영향을 미치지 않는다. 따라서 본 논문에서는 문제를 간단히 하기 위하여 다음과 같은 가정하에 식 (9)와 같이 근사 비용 모델을 사용한다. 즉 FP에서 삭제된 $P(n-1, 2)$ 들은 $P'(n-1, 2)$ 로 갱신된 후 삽입 위치를 포함하는 페이지에 삽입된다. WP에 존재하는 $P(n-1, 2)$ 는 WP내에서 $P'(n-1, 2)$ 로 갱신된다. LP에서 삭제된 $P(n-1, 2)$ 들로 부터 갱신된 $P'(n-1, 2)$ 들과 삽입 위치를 포함하는 페이지에서 삽입 위치 이후에 존재하는 $P(n-1, 2)$ 들은 새로운 페이지에 저장된다.

$$NPIS = PAdiff(n) \times (npi + 1 - Pf) + [1 - PAdiff(n)] \times npis \quad (9)$$

$$npi = \begin{cases} 3, & i=(n-1) \text{이거나 } i=n \text{ 경우 (단 } s \neq 1) \\ 2 \times \lceil GLS/P \rceil, & \text{그 이외의 경우} \end{cases}$$

$$npis = \begin{cases} 0, & XSS \leq P \text{인 경우} \\ npi, & XSS > P \text{인 경우} \end{cases}$$

(iii) 리프 노드 레코드 갱신비용 (CBL)

리프 노드 레코드의 갱신비용 모델은 부리프 노드 레코드의 갱신비용과 동일한 개념으로 구해진다.

VI. 성능 평가 및 고찰

1. 성능 평가 방법

저장비용, 검색비용, 갱신비용, 또한 검색 및 갱신을 위한 전체비용면에서 분리 색인 기법과 참고문헌 [5]에서 제안된 기존 색인 기법의 성능을 평가 비교한다. 우선 참고문헌 [5]에서와 동일한 조건에서 제안된 기법과 기존 기법을 평가 비교하기 위해 경로 길이 2 및 3인 경우의 성능 평가 결과를 제시한다. 또한 경로 길이 4 이상에서 참고문헌 [5]에서 권고하는 경로 분할 방법과의 성능을 비교하기 위해 경로 길이 4 및 5인 경우의 성능 평가 결과를 제시한다.

성능 평가 조건은 참고문헌 [5]에서와 동일하게 클래스 C(1)의 인스턴스 수 $N(1)$ 을 200,000으로 하고 $K(1)$ 의 값은 {1,5,10,50,100}의 값을 사용한다. 이때 참고 공유도 및 경로 길이를 가변시켜가며 성능 비교를 수행한다. 그림 및 표에서 pl 은 경로 길이, uc 는 갱신되는 클래스를 가리킨다.

2. 성능 평가

(1) 저장비용

내포 색인 기법, 분리 색인 기법, 경로 색인 기법의 순으로 작은 저장비용을 가진다. 분리 색인 기법의 경우는 리프 클래스에 대한 정보로서 서로 다른 LID만을 포함하며 서로 다른 LID의 수는 $N(1)/k(1, n-1)$ 로 주어진다.

경로 길이 2의 경우 $K(1)$ 이 증가됨에 따라 서로 다른 LID의 수가 감소되며 따라서 분리색인 기법의 저장비용은 내포 색인 기법의 저장비용과 같게 된다. 경로 색인 기법의 경우는 정보 $P(2, 1)$ 을 가지므로 다른 색인 기법에 비해 2배의 저장비용을 가진다.

경로 길이 3의 경우도 $K(1)$ 이 증가됨에 따라 서로 다른 LID의 수가 감소되며 따라서 분리 색인 기법은 클래스 $C(2)$ 에 대한 IID 저장비용만이 영향을 미쳐 내포 색인 기법의 2배의 저장비용을 가진다. 경로 색인 기법은 경로 정보 $P(3, 1)$ 을 가지므로 인해 내포 색인 기법의 3배의 저장비용을 가진다.

한편 색인 기법들에서 비리프 노드 저장비용은 거의 같은 수준의 비용을 가지므로 그림 3에는 포함시키지 않았다. 또한 내포 색인 기법은 색인 갱신을 위해 각 인스턴스에 후방참조정보를 간직하나 그림 3에서는 이에 대한 저장비용을 포함시키지 않았다. 분리색인 기법 및 경로 색인 기법은 후방참조정보의 저장비용이 요구되지 않는다.

(2) 검색비용

분리 색인 기법 및 내포 색인 기법은 리프 노드 레코드의 페이지 수가 같으므로 동일한 검색비용을 갖는다. 경로 색인 기법은 리프 노드 레코드에 $P(n, 1)$ 단위의 경로 정보를 가짐으로 인해 분리 색인 기법 및 내포 색인 기법에 비해 경로 길이 pl 배의 검색비용을 가진다. 따라서 검색비용면에서 경로 색인 기법은 적합하지 않음을 알 수 있다.

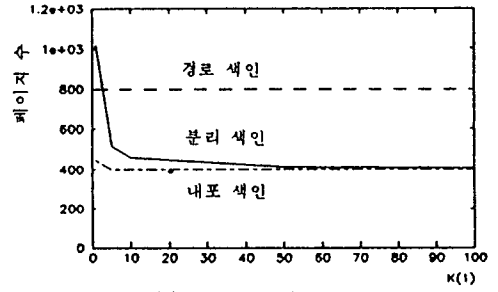
(3) 갱신비용

경로 길이 2 및 3에서 갱신되는 클래스를 루트 클래스로부터 리프 클래스까지 변화시켜가며 갱신비용을 비교한다.

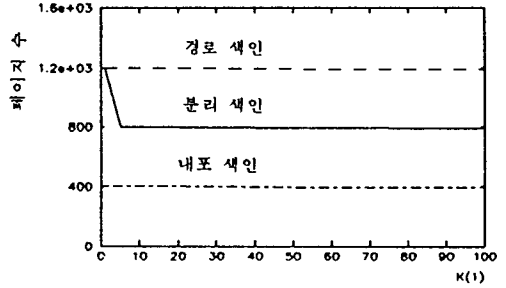
경로 길이 2에서의 갱신비용 평가 결과는 다음과 같다.

그림 5(a)는 갱신되는 클래스가 $C(1)$ 인 경우로서 전방탐색비용은 4로 주어진다. 색인 레코드 갱신시 갱신 대상 RID 및 경로 정보 수는 하나이므로 내포 색인 기법이 가장 작은 갱신비용을 가지며 경로 색인 기법은 내포 색인 기법과 거의 같은 수준의 갱신비용을 가진다. 분리 색인 기법은 다른 색인 기법에 비해 약간 큰 갱신비용을 가진다. 이는 $P(2, 1)$ 을 RID와 LID로 분리하여 레코드를 구성함에 의해 리프 노드 레코드에서 RID 갱신 및 부리프 노드 레코드에서 LID 갱신이 요구되기 때문이다.

그림 5(b)는 갱신되는 클래스가 $C(2)$ 인 경우로서 전



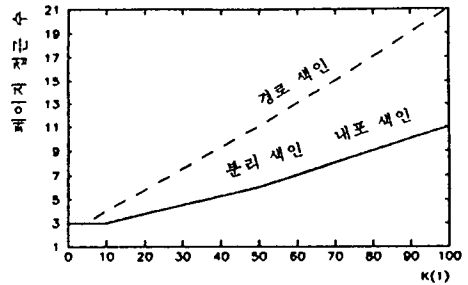
(a) $pl=2, k(2)=50$



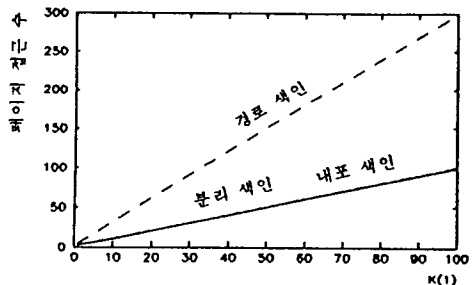
(b) $pl=3, K(2)=50, K(3)=10$

그림 3. 저장비용 비교

Fig. 3. Comparison of storage cost.



(a) $pl=2, k(2)=50$

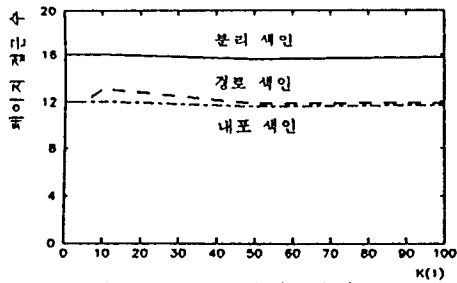


(b) $pl=3, k(2)=50, k(3)=10$

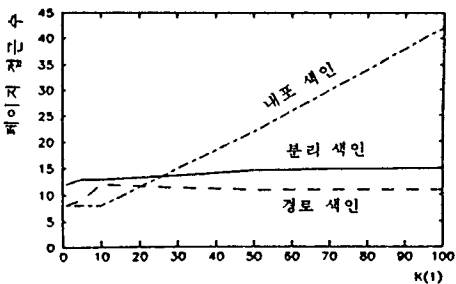
그림 4. 검색비용 비교

Fig. 4. Comparison of retrieval cost.

방탐색비용은 요구되지 않는다. 갱신 대상 RID 및 경로 정보 수는 $K(1)$ 으로 주어지며 내포 색인 기법의 경우 참고 공유도 $K(1)$ 이 증가함에 따라 크게 증가됨을 알 수 있다. 이는 내포 색인 기법의 경우 색인 레코드에 RID만이 존재함으로 인해 RID들은 RID 자체의 올림차순으로만 정렬되어지며 그 결과 갱신 대상 RID의 위치가 색인 레코드내에서 무작위로 존재하기 때문이다. 이로 인해 Yao의 블럭 접근 예측식^[10]에 따라 갱신비용은 갱신되는 RID의 수뿐만이 아니라 색인 레코드의 페이지 수에도 의존하게 된다. 참고문헌 [5]에서는 갱신되는 RID가 한 페이지내에 존재한다는 가정하에 갱신비용을 평가하였기 때문에 이와 같은 점이 고려되지 않았으나 내포 색인 기법의 또 하나의 단점이 되고 있다. 분리 색인 기법은 경로 색인 기법에 비해 약간 큰 갱신비용을 가지며 그림 5(a)의 경우와 동일한 개념으로 설명된다. 분리 색인 기법 및 경로 색인 기법에서는 갱신 대상 정보인 IID들이 언제나 연속적으로 존재한다.



(a) $pl=2, vc=c(1), k(2)=50$



(b) $pl=2, vc=c(2), k(2)=50$

그림 5. 갱신비용 비교 (경로 길이 2)

Fig. 5. Comparison of update cost (path length 2).

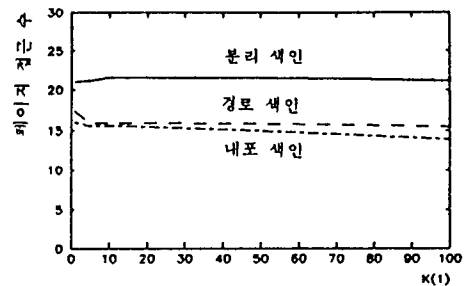
경로 길이 3에서의 갱신비용 평가 결과는 다음과 같다.

그림 6(a)는 갱신되는 클래스가 $C(1)$ 인 경우로서 전방탐색비용은 8로 주어진다. 그림 5(a)의 경우와 같이 갱신 대상 RID 및 경로 정보 수는 하나로 주어지며 그 결과

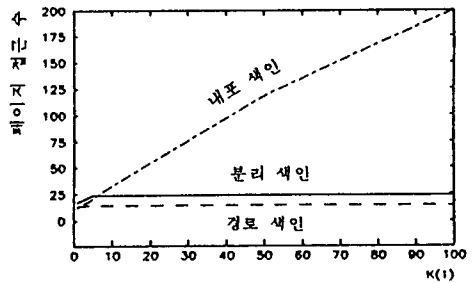
내포 색인 기법, 경로 색인 기법, 분리 색인 기법의 순으로 작은 갱신비용을 갖는다.

그림 6(b)는 갱신되는 클래스가 $C(2)$ 인 경우로서 전방탐색비용은 4로 주어진다. 갱신 대상 RID 및 경로 정보 수는 $K(1)$ 으로 주어진다. $K(1)$ 의 증가에 따라 내포 색인 기법의 갱신비용이 증가됨을 알 수 있다. 이는 갱신 대상 RID의 위치가 무작위로 존재하기 때문이다.

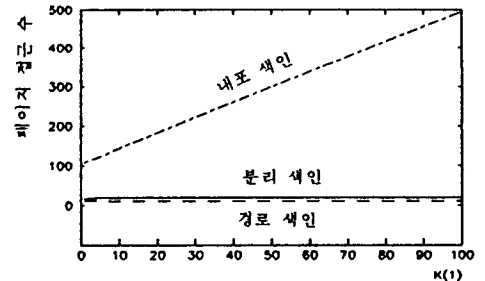
그림 6(c)는 갱신되는 클래스가 리프 클래스인 경우로서 내포 색인 기법의 경우 후방탐색비용이 요구되며 또한 갱신 대상 RID의 무작위 존재 특성으로 인해 가장 큰 갱신비용을 가진다. 분리 색인 기법, 경로 색인 기법에서는 후방탐색비용이 요구되지 않는다.



(a) $pl=3, vc=c(1), k(2)=50, k(3)=10$



(b) $pl=3, vc=C(2), (k)=50, k(3)=10$



(c) $pl=3, vc=C(3), k(2)=50, k(3)=10$

그림 6. 갱신비용 비교 (경로 길이 3)

Fig. 6. Comparison of update cost (path length 3).

(4) 전체비용

그림 7은 경로 길이 2부터 5까지에서 검색 및 갱신을 위한 전체비용을 평가한 것이다. 이때 전체비용은 50%의 검색 및 50%의 갱신이 발생하는 경우를 고려한다. 또한 갱신비용은 주어진 경로에서 각 클래스의 갱신될 확률이 동일한 경우를 고려하여 평균갱신비용을 사용한다. 그림에서 R은 검색비용, AU는 평균갱신비용을 가리킨다. 경로 길이 4 및 5의 경우는 K(1)의 값을 각각 {1, 5, 10, 15, 20, 25, 30} 및 {1, 2, 4, 6, 8, 10}에 대해 평가한다.

경로 길이 2에서는 참고 공유도 K(1)이 50 보다 작은 경우 내포 색인 기법, 경로 색인 기법, 분리 색인 기법의 순으로 작은 전체비용을 가지며 K(1)이 50 보다 큰 경우 분리 색인 기법, 경로 색인 기법, 내포 색인 기법의 순으로 작은 전체비용을 가진다. 이는 K(1)이 증가됨에 따라 내포 색인 기법의 경우 갱신 대상 IID들의 수 및 색인 레코드 페이지 수가 증가하고, 경로 색인 기법의 경우 검색 대상 정보를 갖고 있는 리프 노드 레코드의 페이지 수 증가율이 다른 색인 기법에 비해 높기 때문이다.

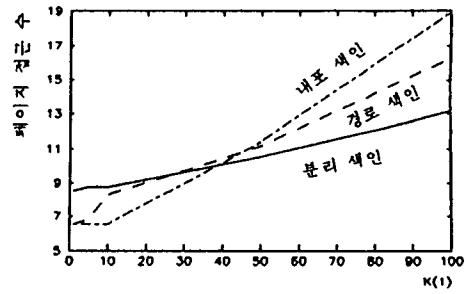
경로 길이 3에서는 분리 색인 기법, 경로 색인 기법, 내포 색인 기법의 순으로 작은 전체비용을 가지며 경로 길이 2에서와 동일한 개념으로 설명된다.

경로 길이 4인 경우는 K(1)이 15 보다 작은 경우 분리 색인 기법, 경로 색인 기법, 내포 색인 기법의 순으로 작은 전체비용을 가지며 K(1)이 15보다 큰 경우 경로 색인 기법의 검색비용 증가율로 인해 내포 색인 기법이 경로 색인 기법 보다 작은 전체비용을 가진다. 경로 길이가 5인 경우도 경로 길이가 4인 경우와 동일한 특성을 가진다.

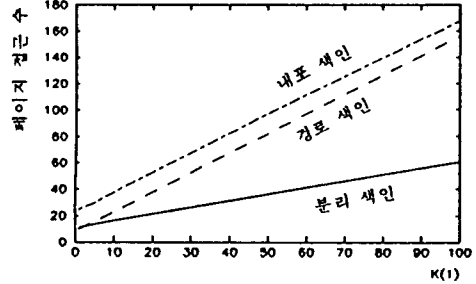
표 2 및 표 3은 전체비용에서 검색비용 및 평균갱신비용의 영향을 분석한 것으로서 경로 길이가 증가함에 따라 내포 색인 기법 및 경로 색인 기법의 경우 각각 평균갱신비용 및 검색비용의 증가로 전체비용이 매우 커짐을 알 수 있다. 또한 각 부경로의 경로 길이가 1인 경로 분할 방법의 의미를 갖는 다중 색인 기법의 전체비용을 표에 포함시켰다. 다중 색인 기법의 경우 가장 작은 갱신비용을 가지나 검색비용이 큼을 알 수 있다.

경로 길이 4 및 5에 대해서는 경로 분할 및 색인 할당 기법을 평가하고 그 결과를 표 3에 포함시켰다. 즉 경로 길이 4 이상의 경로를 경로 길이 3 이하의 부경로들로 분할하고 각 부경로에의 색인 할당 방법은 참고문헌 [5]에서 제시한 방법에 따라 경로 길이가 2 및 3인 경우 각각 내포 색인 및 경로 색인을 할당하였다. 고려된 경로 분할 방법은 표 3에 표시된 바와 같다.

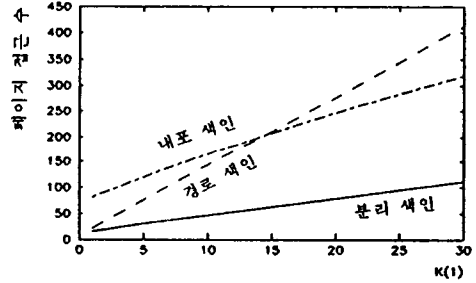
표 3에 나타난 바와 같이 경로 분할 후 색인을 할당하는 방법은 기존의 내포 색인 기법 및 경로 색인 기법에 비해 작은 전체비용을 가진다. 경로 분할 방법은 갱신 범위가 부경로내로 한정됨에 의해 작은 갱신비용을 가진다. 그러나 여러개의 색인을 특입해야함에 의해 큰 검색비용을



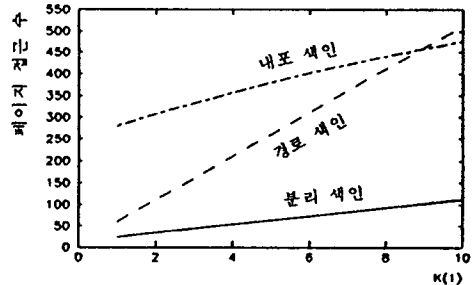
(a) $pl=2, k(2)=50$ (50%R+50%AU)



(b) $pl=3, k(2)=50, k(3)=10$ (50%R+50%AU)



(c) $pl=4, k(2)=15, k(3)=15, k(4)=15$ (50%R+50%AU)



(d) $pl=5, k(2)=10, k(3)=10, k(4)=10$ (50%R+50%AU)

그림 7. 검색 및 갱신을 위한 전체비용 비교
Fig. 7. Comparison of total cost for retrieval and update.

가지게 된다. 특히 색인 구성의 기본 목적은 빠른 검색이라는 점을 고려할때 경로 분할 방법은 역시 내포 질의를 위한 색인 기법으로 적합하지 않음을 알 수 있다. 한편 표 3에 나타난 바와 같이 본 논문에서 제안된 분리 색인 기법

은 내포 색인 기법과 함께 가장 작은 검색비용을 가지며 경로 분할방법에 비해 작은 전체비용을 가짐을 알 수 있다.

표 2. 검색 및 갱신을 위한 전체비용 분석 (경로길이 2 및 3)

Table 2. Analysis of total cost for retrieval and update(path length 2 and 3).

경로 길이	색인기법 비용	분리	내포	경로	다중
	평균갱신 (50%AU)	7.5	8.4	5.6	3.5
	합	10.5	11.4	11.1	29.3
pi=3	검색 (50%R)	25	25	75	161.3
	평균갱신 (50%AU)	10.8	71.9	6.7	3.3
	합	35.8	96.9	81.7	164.6

pi=2; K(1)=50, K(2)=50

pi=3; K(1)=50, K(2)=50, K(3)=10

Ⅶ. 결론

객체 지향 자료 모형의 집약 계층에서 내포 질의 처리를 위한 기존 색인 기법들은 검색비용 및 갱신비용간에 상호 장단점을 가지며 경로 길이가 증가됨에 따라 이들 비용간의 차이가 더욱 크게 나타난다. 본 논문에서는 기존 색인 기법의 이와 같은 점을 고려하여 색인 레코드 구성시 검색 대상 정보와 갱신을 위한 경로 정보를 각 목적에만 사용되도록 분리하여 두개의 레코드로 구성하여 관리하는 분리 색인 기법을 제안하였다. 또한 레코드에서 갱신 대상 정보가 연속적으로 존재하는 경우 갱신 대상 페이지를 분류하여 갱신하는 개선된 갱신 알고리즘을 제시하였다.

제안된 색인 기법에 대한 저장, 검색 및 갱신비용 모델을 제시하였으며 경로 길이 및 참고 공유도를 변경시켜가며 제안된 색인 기법과 기존의 내포 색인 기법 및 경로 색인 기법과의 성능 비교를 수행하였다. 성능 평가 결과 경로 길이 2부터 5까지에서 분리 색인 기법이 기존 색인 기

표 3. 색인기법의 전체비용 분석(경로길이 4 및 5)

(a) 경로길이 4 (b) 경로길이 5

Table 3. Analysis of total cost for retrieval and update(path length 4 and 5).

(a) path length 4, (b) path length 5.

색인기법 비용	색인직접 할당 방법			경로 분할 방법		
	분리	내포	경로	(방법 1)	(방법 2)	다중
검색 (50%R)	50.5	50.5	202	100.3	234.5	241.5
평균갱신 (50%AU)	11.8	156.7	7.9	4.3	6	3.2
합	62.3	207.2	209.9	104.6	240.5	244.7

K(1)=15, K(2)=15, K(3)=15, K(4)=15

(방법 1) P1=C(1)A(1)A(2); 내포 색인
P2=C(3)A(3)A(4); 내포 색인
(방법 2) P1=C(1)A(1); 단일 클래스 색인
P2=C(2)A(2)A(3)A(4); 경로 색인

(a)

색인기법 비용	색인직접 할당 방법			경로 분할 방법		
	분리	내포	경로	(방법 1)	(방법 2)	다중
검색 (50%R)	98.5	98.5	499	198.3	260.4	269
평균갱신 (50%AU)	12.5	376.7	10.3	5.8	3.7	3.2
합	111	475.2	509.3	204.1	264.1	272.2

K(1)=10, K(2)=10, K(3)=10, K(4)=10, K(5)=10

(방법 1) P1=C(1)A(1)A(2); 내포 색인
P2=C(3)A(3)A(4)A(5); 경로 색인
(방법 2) P1=C(1)A(1); 단일 클래스 색인
P2=C(2)A(2)A(3); 내포 색인
P3=C(4)A(4)A(5); 내포 색인

(b)

법에 비해 검색 및 갱신을 위한 전체비용면에서 작은 비용을 가짐을 보였다. 경로 길이 4 이상에서는 경로 분할 후 색인을 할당하는 방법에 대해서도 평가 비교하였으며 그 결과 경로 분할 방법에 비해서도 분리 색인 기법이 더 작은 전체비용을 가짐을 보였다.

향후 VLSI 설계 자료, 의료 화학 정보 및 문서 정보등과 같은 다중 매체 정보 관리 및 검색 시스템에 본 연구에서 제시한 색인 기법의 적용 연구가 요구되며 더 나아가서는 색인 구축시 질의 빈도를 고려한 경로 선택 문제 및 질의 최적화에 대한 연구가 요구된다.

參 考 文 獻

- [1] J. Benerjee, H.T. Chou, J. Garza, W. Kim, D. Woelk, N. Ballou and H.J. Kim; "Data model issues for object-oriented applications," *ACM Trans. on Office Information Systems*, vol. 5, no 1, pp. 3-26, Jan. 1987.
- [2] D. Woelk, W. Kim, W. Luther, "An object-oriented approach to multimedia databases," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 311-325, May 1986.
- [3] W. Kim, "A model of queries for object-oriented databases," *Proc. 15th Int'l Conf. on VLDB*, Amsterdam, pp. 423-432, Aug. 1989.
- [4] D. Maier and J. Stein, "Indexing in an object-oriented DBMS," *Proc. Int'l Workshop Object-Oriented Database Systems, Languages*, Pacific Grove, CA, pp. 171-182, Sep. 1986.
- [5] E. Bertino, W. Kim, "Indexing techniques for queries on nested objects," *IEEE trans. on Knowledge and Data Eng.*, vol. 1, no. 2, pp. 196-214, June 1989.
- [6] W. Kim, F. Lochovsky, *Object-Oriented Concepts, Applications, and Databases*, pp. 371-394, Reading, MA: Addison-Wesley, 1989.
- [7] F. Fotouhi, T.G. Lee, and W.I. Grosky, "The generalized index model for object-oriented database systems," *10th Annual Int'l Phoenix Conf. on Computers and Communications*, pp. 302-308, Mar. 1991.
- [8] E. Bertino, "An Indexing technique for object-oriented databases," *Proc. 7th Int'l Conf. on Data Eng.*, pp. 160-170, Apr. 1991.
- [9] D. Comer, "The ubiquitous B-tree," *ACM Comput. Surveys*, vol. 11, no. 2, pp. 121-137, June 1979.
- [10] S.B. Yao, "Approximating block accesses in database organization," *Communications of the ACM.*, vol. 20, no. 4, pp. 260-261, Apr. 1977.

 著 者 紹 介

權 用 武 (正會員) 第 27 卷 第 6 號 參照
 현재 한양대학교 전자공학과
 박사과정

朴 容 震 (正會員) 第 29 卷 B編 第 4 號 參照
 현재 한양대학교 전자공학과
 교수