김일성종합대학학보
(자연과학)
주체103(2014)년 제60권 제6호

JOURNAL OF **KIM IL SUNG** UNIVERSITY
(NATURAL SCIENCE)
Vol. 60 No. 6 JUCHE103(2014).

비트단위의 열쇠색인을 리용한 BURP-Trie 구조에 대한 연구

김대명, 조은철

1. 자료기지관리체계에서 리용되는 Trie와 B-나무의 주요특징과 제한성

Trie구조는 자료의 삽입과 삭제, 갱신이 일어나지 않는 정적인 자료구조에서 많이 리용되는 고속탐색알고리듬으로서 다음과 같은 우점을 가진다.

① 열쇠탐색의 시간복잡도는 열쇠모임의 크기에 관계없이 $O(1)$이다.[1, 3]

② 열쇠삽입시간은 열쇠모임의 크기에 선형으로 비례한다.[2]

③ 열쇠삭제시간은 열쇠의 길이에만 비례하므로 $O(1)$의 시간복잡도를 가진다.[2]

Trie를 표현하기 위한 자료구조에서 가장 대표적인것은 2중배렬구조에 Tail배렬을 리용한 2중배렬압축TailTrie로서 다음과 같은 일련의 결함을 가지고있다.

① Trie에 구축할수 있는 열쇠모임이 제한된다.

② 삽입계산복잡도가 열쇠모임의 크기에 따라 폭발적으로 증가한다.

③ 범위탐색을 B－나무에 비하여 효과적으로 지원할수 없는것이다.

④ Trie나무를 주기억에 한번에 올려놓을수 없는 경우 페지화를 진행하지 않으므로 디스크입출력비용이 높아지게 된다.

한편 자료기지관리체계에서 많이 리용되는 B－나무구조는 내부마디들이 탐색경로를 유도하고 말단마디들이 자료입력점을 가지는 균형나무(balanced tree)이다.

B－나무의 주요특징은 다음과 같다.

① 나무에서 연산(삽입, 삭제)을 수행해도 나무의 균형은 유지된다.[1, 5]

② 뿌리를 제외한 각 마디는 적어도 50%이상 차게 된다. 그러나 화일은 일반적으로 축소되기보다는 확장되므로 삭제할 때에는 나무를 다시 조정하지 않고 단순히 찾아서 제거만 한다.[4]

③ 레코드를 탐색할 때에는 뿌리로부터 알맞는 말단까지 가기만 하면 된다.

B－나무는 색인입구점을 저장하기 위한 공간부하가 있지만 대신에 정렬화일의 우점과 효과적인 삽입, 삭제알고리듬의 우점을 다 가진다. 그러나 B－나무는 하쉬구조나 Trie에 비해 동등탐색속도가 현저하게 떨어지는 결함이 있다.

론문에서는 Trie와 B－나무의 우결함을 객관적으로 분석한데 기초하여 자료기지의 색인구축 및 탐색속도를 보다 높이며 대용량자료기지들을 효과적으로 관리할수 있도록 하기 위하여 가능한 모든 자료형들을 열쇠로 리용할수 있도록 하며 효과적인 열쇠압축을 지원할수 있는 비트단위의 열쇠검색을 리용한 개선된 Trie색인구조를 제안하였다.

2. BURP-Trie색인구조의 정의

정의 1 열쇠의 내부표현값에 따르는 나무의 자식이행정보를 담고있는 최소단위를 슬로트라고 하고 다음과 같이 표시한다.

Slot=<page_num, node_num>

슬로트들의 구성요소들의 의미와 길이는 표 1과 같다.

표 1. Slot의 구성요소의 의미 및 길이

번호	성원이름	의미	길이
1	page_num	자식마디의 페지번호	4B
2	node_num	해당 페지안에서의 마디번호 (해당 페지안에서의 변위)	2B

정의 2 모든 열쇠탐색의 시작으로 되는 마디를 뿌리마디라고 부르고 다음과 같이 표시한다.

R=<slot(0), slot(1)>

이러한 뿌리마디는 하나이다.

slot(i)(i=0, 1)는 열쇠의 첫 비트의 내부표현값으로부터 이행하여야 할 자식이행정보 즉 자식마디에로의 지적자를 담고있으며 이것을 따라가면 계산된 내부표현값에 해당한 비트를 첫 비트로 가지는 모든 열쇠탐색을 위한 슬로트에 도달할수 있다.

정의 3 뿌리마디를 제외한 열쇠탐색에 리용되는 나무의 중간에 있는 마디를 내부마디라고 부르고 다음과 같이 표시한다.

M=<Hm, slot(#), slot(0), slot(1)>

여기서 slot(#)는 내부마디에 고유한 슬로트로서 해당 마디까지의 비트렬로 끝나는 열쇠를 탐색하기 위한 이행정보를 담고있으며 그 구조는 slot(i)와 같다. 그리고 Hm은 내부마디머리부로서 그 구조는 표 2와 같다.

표 2. 내부마디머리부구조

번호	성원이름	의미	길이
1	Parent_slot	어미마디를 가리키는 슬로트로서 열쇠삭제때 나무를 재구축하기 위한 정보를 담는다.	6B
2	Prunded_flag	어미마디와 자기사이에 열쇠압축이 진행되였는가를 가리키는 기발	1bit
3	Prunded_len	압축된 열쇠길이	7bit
4	Prunded_String	압축된 열쇠비트렬	8B

정의 4 실제의 <열쇠, 자료>쌍의 레코드들을 포함하고있는 마디를 잎마디라고 부르고 다음과 같이 표시한다.

L=<K, D>

여기서 K는 열쇠이고 D는 열쇠 K에 대응하는 자료이다.

이때 <열쇠, 자료>쌍의 길이가 고정되여있지 않고 가변적일수 있기때문에 1개의 마디에 들어갈수 있는 레코드개수는 고정되여있지 않게 된다.

또한 <열쇠, 자료>쌍의 길이가 1개의 마디용량보다 큰 경우에는 잎마디를 새로 추가하여 련결하는것이 아니라 <열쇠, 자료>쌍의 나머지부분을 저장하기 위한 보조적인 마디들을 리용하게 하며 이것을 넘침마디라고 부르고 L0으로 표시한다.

넘침마디들은 해당한 <열쇠, 자료>쌍의 앞부분이 들어있는 잎마디와 련결목록으로 련결된다.

잎마디의 매 <열쇠, 자료>쌍부분을 확대해 보면 그림 1과 같다.

잎마디안에 놓이는 <열쇠, 자료>쌍들은 정렬순서상 열쇠크기에 따라 정렬되여 놓인다.

또한 잎마디들은 2중련결목록에 의하여 서로 련결되여있어 잎마디들사이에서의 열쇠크기 순서를 그대로 유지하도록 한다.

그림 1. <열쇠, 자료>쌍의 구조

B－나무에서와 마찬가지로 잎페지들에 놓이는 련결목록의 순서를 고려하면 모든 레코드들은 해당한 열쇠의 크기순서로 정렬되여 저장되도록 할수 있으며 이것은 범위탐색도 효과적으로 지원할수 있다.

이상의 개념들을 정의한데 기초하여 BURP-Trie를 다음과 같이 정의한다.

정의 5 R를 뿌리마디, M을 내부마디들의 유한모임, L을 잎마디들의 유한모임(넘침마디포함), g를 마디이행규칙이라고 할 때 이 요소들의 쌍 (R, M, L, g)를 BURP-Trie라고 부르고 다음과 같이 표시한다.

$$TP = (R, M, L, g)$$

한편 마디이행규칙 g는 다음과 같이 정의한다.

정의 6 탐색하려는 열쇠를 $K = k1k2\cdots kn$ (여기서 ki는 열쇠 K의 i번째 비트), 내부표현값을 $vi = val(ki)$라고 할 때 상태이행함수

$$g:\{vi \mid i \in \{\#,\ 0,\ 1\}\} \times \{R,\ M\} \Rightarrow \{M,\ L\}$$

을 마디이행규칙이라고 부른다.

ui와 uk가 뿌리마디 혹은 내부마디라면 $g(vi, uj) = uk$로 되는 uj와 uk가 존재할 때 이 두 마디는 부모자식관계에 있다고 말하며 여기서 uj는 부모마디, uk는 자식마디라고 부른다.

g는 열쇠 K의 매 비트의 내부표현값 vi에 따르는 마디안의 슬로트를 찾아서 자식마디의 페지번호와 마디번호를 얻는다.

BURP-Trie의 개념적구조는 그림 2와 같다.

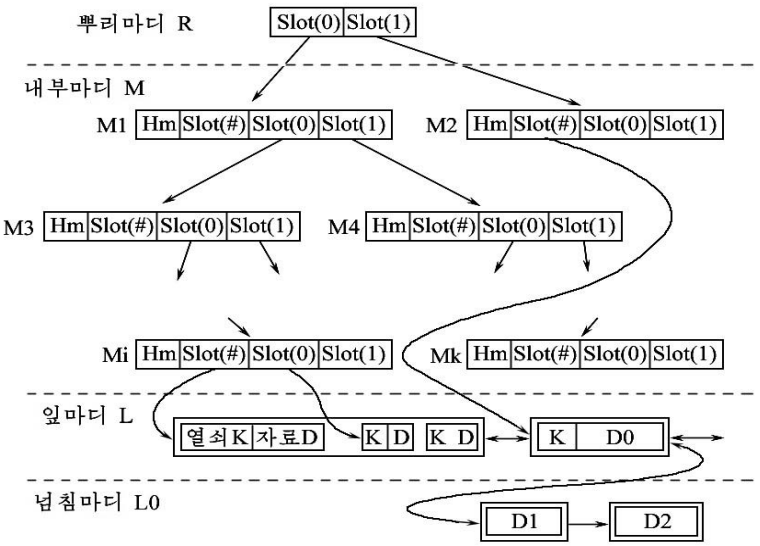

그림 2. BURP-Trie의 개념적구조

3. BURP-Trie색인구조의 특징

BURP-Trie의 정의로부터 그것의 특징을 다음과 같이 고찰할수 있다.

① Trie색인구조로서의 특징을 가지고있는것으로 하여 탐색속도에서 Trie와 같으면서도 보다 우월한 색인성능을 가진다. 즉 탐색에서는 비교연산이 없고 단지 대입연산만 있으며 따라서 탐색시간은 원래의 Trie에서와 같다.

② 열쇠단어의 해당한 바이트가 들어갈 공간이 미리 할당되여있으므로 열쇠삽입때 나무의 뿌리마디나 내부마디에서 나무조정이 일어나지 않는다. 다만 열쇠압축을 진행한 경우 그 슬로트를 끝슬로트로 하는 열쇠나 그 마디까지의 내부표현값들이 같은 열쇠를 삽입할 때 새 마디를 할당하여야 한다. 따라서 열쇠삽입속도가 TailTrie에 비하여 빠르게 된다.

③ 색인나무가 페지화되여있으므로 B－나무의 우점을 그대로 가지고있다.

마디의 용량이 고정인것으로 하여 마디를 페지에 정적으로 배치할수 있으며 따라서 탐색때 필요한 페지만을 주기억에 적재하면 되므로 디스크입출력회수를 B－나무에 맞먹을 정도로 줄일수 있다.

④ 열쇠압축을 리용하므로 열쇠단어를 이루는 바이트들의 출현빈도에 따라서 B－나무보다 더 적은 디스크입출력을 요구할수도 있다.

⑤ 잎마디와 넘침마디에서는 B－나무의 방식을 그대로 리용하였으므로 범위탐색때 나무를 따라 다시 올라가야 하던 Trie의 약점을 완전히 극복하고 B－나무와 같은 성능을 보장할수 있다.

맺 는 말

B－나무와 Trie의 우점을 모두 살리면서도 그들이 가지고있던 결함을 극복할수 있도록 새롭게 BURP-Trie색인구조를 제안하였다.

제안한 색인구조는 색인구축 및 탐색에서 B－나무나 Trie를 모두 릉가하는 우월한 색인구조로 된다.

참 고 문 헌

[1] H. Shang; Trie Methods for Text and Spatial Data on Secondary Storage, Mc-Gill University, 70～95, 2001.

[2] D. E. Zegour; Elsevier Information and Software Technology, **46**, 923, 2004.

[3] Minsoo Jung et al.; Information Processing and Management of Elsevier, **38**, 221, 2002.

[4] J. Bourdon; Theoretical Informatics and Applications, **35**, 163, 2002.

[5] M. E. Nebel; Theoretical Computer Science, **270**, 441, 2002.

주체103(2014)년 2월 5일 원고접수

BURP-Trie Indexing Structure using bit-unit Keyword Index

Kim Tae Myong, Jo Un Chol

We researched about indexing structure using for several database and construction of dictionary including database management systems.

In this paper we have proposed improved Trie structure, Bit-unit Retrieval Paged Trie (BURP-Trie) that used paged indexing structure and keyword index based bit-unit.

This indexing structrure has better performance than B-tree and Trie for data indexing and retrieval.

Key words: data indexing, indexing structrure