'97 International Conference Multimedia Databases on Internet
October 10, 1997, Seoul Korea

속성 중복을 고려한 릴레이션의 수직 분할방법

Vertical Relation Partitioning Method With Attribute Replications

유종찬(한양대학교)

김재련(한양대학교)

Abstract

관계형 데이터베이스의 성능을 향상시키는데 중요한 요소 중의 하나는 트랜잭션을 처리하기 위해 데이터를 디스크에서 주 기억장치로 옮기는데 필요한 디스크 액세스(access) 횟수이다. 본 연구는 관계형 데이터베이스에서 트랜잭션을 처리할 때, 릴레이션(relation)을 수직분할하여 디스크에 단편(fragment)으로 저장하므로써 필요한 단편만 액세스하여 액세스 횟수를 감소시키는데 목적이 있다.

단편에 속성을 중복할당하여 수직분할하므로써 트랜잭션을 만족시키는 단편의 수를 감소시켜 중복할당을 고려하지 않은 방법보다 디스크 액세스 횟수를 감소시킬 수 있다. 갱신트랜잭션의 경우 하나의 속성이 갱신되면 중복된 속성을 모두 갱신하여야 하므로 액세스 횟수가 증가하지만, 조회트랜잭션의 경우 각 단편에 속성을 중복할당하여 액세스 횟수를 감소시킬 수 있다. 본 연구에서는 속성의 중복을 허용하여 단편을 구성하는 경우에 중복을 고려하지 않은 경우를 포함하므로 효과적으로 디스크 액세스 횟수를 감소시킬 수 있다.

본 연구에서는 중복할당을 고려하여 디스크의 액세스 횟수를 최소화시킬 수 있는 수직분할문제의 0-1 정수계획모형을 개발하고, 모형에 대한 최적해법으로 분지한계법을 제안한다.

1. 서 론

1.1 연구의 배경과 목적

관계형 데이터베이스에서 트랜잭션을 처리할 때 주기억 장치와 디스크간에 이동해야하는 데이터의 양이 많으면 디스크의 액세스 횟수가 증가하게되어 그 만큼 트랜잭션에 대한 반응시간이 증가하게 된다. 그러나 트랜잭션은 대상 릴레이션의 모든 속성을 필요로 하지 않은 경우가 대부분이다. 그러므로 릴레이션을 몇 개의 단편으로 수직분할하여 불필요한 데이터의 이동 양을 줄이면 디스크의 액세스 횟수를 줄일 수 있다. 본 연구는 트랜잭션의 반응시간은 디스크 액세스 횟수에 비례하므로 데이터베이스의 물리적 설계를 실시할 때 속성을 중복하여 수직분할하므로써 데이터베이스의 성능을 향상시키는 방법을 연구한다.

1.2 기존 연구

수직분할에 대한 기존 연구로 Hoffer와 Severance[6]는 속성들을 친밀도가 높은 속성들로 그룹핑하는 알고리듬을 개발하였고, navathe[9]는 [6]의 연구를 확장하여 2단계 이진 수직분할 알고리듬을 제안 하였다. navathe 와 Ra[10]는 [9]의 수직분할방법의 친밀도 행렬에서 두 속성들 사이의 친밀도를 edge값으로 하는 친밀도 그래프 알고리듬을 개발하였다. 그러나 위의 연구에서는 구체적인 물리적 액세스 비용을 고려하지 않았다. Cornell과 Yu[4,5]는 속성들을 물리적인 단편에 할당하여 디스크 액세스 횟수를 최소화하는 이진분할 선형 정수계획법을 개발하였다. 또한 재분할에 관한 수리모형을 제안하였으나, 분할의 수가 커지면 수리적 접근 방식이 불가능하다. Chu[2]는 Cornell and Yu의 문제를 트랜잭션에 근거한 접근 방법으로 이진 수직분할을 연구하였다. Yoon[1]은 디스크액세스 횟수를 줄이기 위한 수직분할문제의 0-1정수 계획모형을 개발하고, 모형의 해법으로 분지한계법을 제안하였으나 속성의 중복을 고려하지 않았다. 본 연구에서는 속성중복을 허용하여 속성중복을 고려하지 않은 방법보다 액세스 횟수를 감소시킬 수 있는 0-1정수 계획모형을 개발하고 해법으로 N-ary 분지한계법을 제안한다.

2. 수직분할 문제의 개요

수직분할 문제는 릴레이션의 속성들을 몇 개의 단편에 할당하여 가능한 조합들 중에서 최적 조합을 찾는 문제이다. 중복이 없는 경우에 a개의 속성을 서로 다르게 수직분할하는 경우의 계산량은 $O(a^a)$이 되고, 중복을 고려하는 경우에 단편의 수가 f개일 때 계산량은 $O((2^f-1)^a)$이 되는 NP -Complete문제가 된다. 디스크의 액세스 횟수는 트랜잭션의 액세스 패턴, 액세스 빈도수, 액세스 방법에 따라 결정된다. 관계형 데이터베이스에서는 액세스방법에 따라 액세스 횟수가

다르고 각 스캔방법들에 대해 액세스 횟수를 추정해 보면 다음과 같다.

Scan Method Type

• *Segment scan method* # of access = $\frac{\text{(\# of tuple)*(length of tuple)}}{\text{(page size)*(prefetch blocking factor)}}$
• *Clustered index scan method* # of access = $\frac{\text{(\# of tuple)*(selectivity)*(length of tuple)}}{\text{(page size)}}$
• *Unclustered index scan method* # of access = (# of tuple)*(selectivity)

위의 식에서, 액세스 횟수는 segment scan방법과 clusterd index scan 방법에서 터플의 길이에 비례하므로, 속성을 단편에 배치하는 방법에 따라서 릴레이션을 스캔할때 액세스횟수를 줄일 수 있다. 이러한 점에 착안 하여 속성중복을 허용하여 액세스 횟수를 최소화 시킬 수 있는 방법을 제안한다.

3. 중복을 허용한 수직분할모형

본 연구에 사용되는 기호를 정의하고, 중복허용 수직분할문제의 수리모형을 제안한다

3.1 사용 기호

본 연구에서 사용하는 기호와 입력변수 및 결정변수는 다음과 같다

*** 사용기호**

a : 릴레이션의 속성 수

t : 릴레이션을 액세스하는 트랜잭션의 수

f : 단편의 개수

j : 속성 번호 (j = 1, ... , a)

i : 트랜잭션 번호 (i = 1, ... , t)

l : 단편 번호 (l = 1, ... , f)

R_t : 조회 트랜잭션의 집합

U_t : 갱신 트랜잭션의 집합

N_l : 단편 l의 터플 길이 (byte)

R_i : 조회 트랜잭션 i의 액세스 비용

U_i : 갱신 트랜잭션 i의 액세스 비용

*** 입력변수**

IDL : 키 속성의 길이 (byte)

ATL_j : 속성 j 의 길이

CDL : 릴레이션의 터플의 수

$Freq_i$: 트랜잭션 i의 단위 시간당 발생 횟수

SEL_i : 트랜잭션 i의 명제 만족율

$$AUM_{ij} = \begin{cases} 1 & \text{트랜잭션 i 가 속성 j를 사용하는 경우} \\ 0 & \text{그렇지 않은 경우} \end{cases}$$

$$COM_{jl} = \begin{cases} 1 & \text{단편l이 트랜잭션i가 요구하는 속성j를 최소로 만족시키는 단편들을 포함하는 경우} \\ 0 & \text{그렇지 않은 경우} \end{cases}$$

UTR_{il} = { 1 갱신트랜잭션 i가 단편 l을 필요로 하는 경우
0 그렇지 않은 경우

RTR_{il} = { 1 조회트랜잭션 i가 단편 l을 필요로 하는 경우
0 그렇지 않은 경우

*** 결정변수**

X_{jl} = { 1 속성 j가 단편 l에 할당될 경우
0 그렇지 않은 경우

3.2 수리 모형

중복허용 수직분할 문제의 수리모형은 다음과 같다.

Min

$$TC = \sum_{i \in R_t} R_i \cdot Freq_i + \sum_{i \in U_t} U_i \cdot Freq_i \qquad (3.1)$$

Subject to

$$R_i = \sum_{l=1}^{f} \{ CDL \cdot SEL_i \cdot (N_l + IDL) \} \cdot RTR_{il} \qquad \forall\ i \qquad (3.2)$$

$$U_i = \sum_{l=1}^{f} \{ CDL \cdot SEL_i \cdot (N_l + IDL) \} \cdot UTR_{il} \qquad \forall\ i \qquad (3.3)$$

$$N_l = \sum_{j=1}^{a} ATL_j \, X_{jl} \qquad \forall\ l \qquad (3.4)$$

$$1 \le \sum_{l=1}^{f} X_{jl} \le J \qquad \forall\ j \qquad (3.5)$$

$$RTR_{il} = \begin{cases} 1 & \sum_{j=1}^{a} AUM_{ij} \cdot X_{jl} \cdot COM_{ji} > 0 \\ 0 & \sum_{j=1}^{a} AUM_{ij} \cdot X_{ji} = 0 \end{cases} \quad \forall\ i, l \quad (3.\ 6)$$

$$UTR_{il} = \begin{cases} 1 & \sum_{j=1}^{a} AUM_{ij} \cdot X_{ji} > 0 \\ 0 & \sum_{j=1}^{a} AUM_{ij} \cdot X_{ji} = 0 \end{cases} \quad \forall\ i, l \quad (3.\ 7)$$

$$X_{jl}=[0,\ 1],\quad UTR_{il}=[0,\ 1],\quad RTR_{il}=[0,\ 1],\quad COM_{jl}=[0,\ 1] \quad \forall\ i, j, l \quad (3.\ 8)$$

목적식 (3. 1)은 모든 트랜잭션의 액세스 비용의 합이다. 조회트랜잭션과 갱신트랜잭션으로 구성되고, 각 트랜잭션이 접근해야 하는 데이터의 양이 감소하면 총 액세스 비용은 감소한다. 제약식 (3. 2)는 조회트랜잭션 i의 액세스 비용을 나타낸다. 제약식 (3. 3)은 갱신트랜잭션 i의 액세스 비용을 나타낸다. 제약식 (3. 4)는 단편 l의 터플 길이를 나타낸다. 식 (3. 5)은 각 단편에 포함시킬 수 있는 속성의 수를 나타내는 식으로 본 연구에는 중복할당을 고려하므로 단편의 수만큼 속성을 할당하는 것이 가능하다. 식 (3. 6)은 조회트랜잭션 i가 단편l에 할당된 속성들 중에서 적어도 하나의 속성을 필요로 하면 단편l을 액세스 해야하는 조건으로 본 연구에서는 트랜잭션을 만족시키는 단편의 조합이 여러개가 있을 수 있다. 이때 변수 COM_{jl}에 의해서 조회트랜잭션 i를 최소로 만족시키는 단편의 조합을 찾는다. 식 (3. 7)은 갱신트랜잭션 i가 단편 l에 할당된 속성들 중에서 적어도 하나의 속성을 필요로 하면, 단편 l을 액세스 해야 하는 조건이다. 식 (3. 9)는 변수가 0 또는 1의 값을 갖는 제약이다.

4. 분지한계법을 이용한 해법

본 장에서는 중복허용 수직분할문제의 최적 해법을 제안한다. 또한 속성의 수를 감소시켜 알고리듬의 계산시간을 줄일 수 있는 초기처리 방안을 제안하고 마지막으로 본 연구에서 제안하는 분지한계법을 이용한 알고리듬의 효율을 평가한다.

4.1 분지한계법

속성을 단편에 추가해서 모든 속성을 할당하고, 중복도 허용된 하나의 가능해

(feasible solution)을 구한 후 다른 조합으로 생성되어지는 열등해를 제외시킬 수 있다. 이렇게 속성 할당의 모든 경우에 대해 가능한 경우를 변경시켜 가면서 최적해를 찾는 분지한계법의 알고리듬 단계(Algorithm Step)는 다음과 같다.

* ***Algorithm Step***

[단계 0] 초기화

- 입력변수 AUM_{ji}, ATL_j , $Freq_i$, SEL_i, CDL, IDL, f를 결정한다.
- f에 의해 결정되는 모든 가능해의 나무를 구성한다.
- 각 노드의 비용 Cal_N_C=0으로 한다.
- 목적식의 값 TC = ∞ 로 한다.

[단계 1] 분지

- 깊이 우선순위로 가지치기 하지않은첫번째 리프노드(leaf node)를 선택한다.
- 선택된 후보노드의 Cal_N_C를 구한다.

[단계 2] 깊이(depth)가 같은 노드의 탐색

- 같은 부모노드를 갖는 이웃한 자식노드로 이동하여 Cal_N_C를 구한다.

[단계 3] 한계

- 만약 선택된 노드가 리프노드이고, Cal_N_C가 TC 보다 작으면 TC 의 값을 Cal_N_C로 바꾼다.

[단계 4] 깊이(depth)가 다른 노드의 탐색

- 만약 같은 부모노드를 갖는 자식노드가 없으면, 부모노드의 이웃노드로 이동하여 Cal_N_C를 구한다.

[단계 5] 가지치기

- 만약 Cal_N_C가 TC 보다 크면, 그 노드의 자식 노드를 가지 친다.

그렇지 않으면, 그 노드의 자식노드를 탐색한다.

[단계 6] 종료

- 가능해 나무에서 가지치기 안된 남아있는 노드가 없으면 종료하고,현재의 TC 값을 최적해로 한다.
  그렇지 않으면 [단계 2]로 간다.

4.2 분지한계법에서의 비용계산 방법

분지한계법에 사용되는 속성에 대한 용어의 정의와 비용계산 알고리듬 문제의 특성은 다음과 같다.

- 1차배치 속성 : 단편을 구성할 때 첫 번째로 배치된 속성으로 아래의 단편 구성행렬에서 원으로 표시한 속성

- 중복배치 속성 : 1차배치속성이 아니고, 중복하여 배치된 모든 속성

Fragment Composition Matrix

	속성1	속성2	속성3	속성4	속성5	속성6
단편 1	①		①			
단편 2	1	①			①	
단편 3	1	1	1	①	1	
단편 4		1				①

- 비용계산 알고리듬의 특성

① 단편에 1차배치 속성을 추가하면 액세스 비용은 증가한다.

② 단편에 1차배치 속성이 포함되지 않으면 중복배치 속성을 할당할 수 없다.

4.3 초기처리 방법

본 연구에서 제안한 모형의 해법은 속성의 수에 따라 크게 영향을 받는다. 그러므로 분지한계법에서 고려해야 할 속성의 수를 줄이면 변수의 수가 줄어들어 계산량을 줄일 수 있다. 그러므로 클러스터링 기법중의 하나인 CI알고리듬을 이용하여 본 연구에서 제안한 알고리듬의 초기에 적용하므로써 동일한 트랜잭션을

처리하는 속성을 찾아낼 수 있다. 이러한 속성들을 하나의 단편에 할당하여 계산하면, 적은 계산량으로 해를 찾을 수 있다.[1]

4.4 알고리듬의 평가

분지한계법의 알고리듬을 C/C++로 구현하여 Pentium(150Mhz)컴퓨터로 속성이 5개, 트랜잭션이 5개인 속성사용 행렬에 대해 다음 Table에 따라 20문제를 랜덤하게 만들어 비중복, 중복허용 알고리듬에 대한 비교실험을 하였다.

Parameter values in the experimental example

Parameters	Values
Identifier Length	IDL = 4
Attribute Length	$1 \leq ATL_j \leq 35$
Transaction Frequency	$1 \leq Freq_i \leq 65$
Transaction Type	Retrieval
Selectivity of transaction	$SEL_i = 1$
A probability(Pr) of an attribute accessed by a transaction	0.4 or 0.6
Cardinality	CDL = 10,000

실험결과는 다음과 같다.

Comparison with unreplicted algorithm and replicated algorithm

Algorithms	Avg. Cost Reduction(%)			Max. Cost Reduction (%)	Min. Cost Reduction (%)
	Pr=0.4	Pr=0.6	Total		
비중복 분지한계법	51.27	31.22	41.25	60.65	21.82
중복허용 분지한계법	62.38	45.18	53.78	71.10	33.74

위의 표에서 알수 있듯이 릴레이션을 수직분할하지 않았을 경우와 비교해서 비중복 분지한계법에서는 비용절감이 평균 41.25%였고, 중복허용 분지한계법에서는 53.78%였다. 그러므로 속성의 중복을 허용한 분지한계법이 비중복 분지한계법 보다 비용면에서 12.5%가 감소됨을 보여주고 있고, Pr값이 작을수록 비용

절감효과가 크다는 것을 알 수 있다. 또한 문제에 따라서는 속성중복을 하므로써 비용을 감소시키지 못하는 경우에는 비중복 분지한계법과 동일한 해를 산출한다. 속성중복 허용 분지한계법에서는 비중복문제의 최적해를 포함하기 때문에 중복 허용 알고리듬의 최적해는 반드시 비중복 알고리듬의 최적해 보다 비용이 작거나 같다. 위의 예제를 푸는데 평균 계산시간은 937.57초가 걸렸으며, 속성의 수가 많지 않은 문제에 본 연구에서 제안한 알고리듬을 적용하면 효율적으로 최적해를 찾을 수 있다.

5. 결 론

본 연구에서는 디스크 액세스 횟수를 최소화하는 속성중복허용 0-1 정수 계획 모형을 개발하고, 이 모형에 대한 해법으로 분지한계법을 제안하였다. 또한 동일한 트랜잭션을 처리하는 속성을 찾아내어 문제의 크기를 줄일 수 있는 초기처리 방법을 제안하였다.

관계형 데이터베이스의 성능은 디스크 액세스 시간에 따라 크게 영향을 받는다. 트랜잭션이 릴레이션의 모든 속성을 다 필요로 하지 않으므로 이러한 특성을 이용하여 트랜잭션은 가능한 적은 수의 단편만을 액세스 하도록 한다. 속성중복을 고려하지 않은 경우보다 속성의 중복을 허용하여 구한 최적해가 비용이 적은 것으로 나타났다.

속성의 수가 많아지면 알고리듬의 계산량이 지수적으로 증가한다. 그러므로 본 알고리듬의 최적해에 근접하는 해를 짧은 시간에 찾을 수 있는 발견적 기법의 개발이 요구된다.

참 고 문 헌

[1] 윤병익, 김재련, "데이터베이스의 물리적설계에서 분지한계법을 이용한 N-ary수직분할문제, " *대한산업공학회*, 제22권, 4호, pp. 567-578,1996

[2] Chu, W. W and I. T. Ieong, "A transaction-based approach to vertical partitioning for relational database systems, " *IEEE Trans. on Software.*, Vol.19, No.8, pp.804-812, 1993

[3] Cheng, C.,"Algorithms for vertical partitioning in database physical design, "*Omega, Int.J. Mgnt. Sci.*, Vol.22, No.3, pp.291-303, 1994

[4] Cornell, D. W. and P. S. Yu, " An effective approach to vertical partitioning for physical design of relational database," *IEEE Trans. Software Eng.*, Vol.16, No.2, pp. 248-258, 1990.

[5] Cornell, D. W. and P. S. Yu, " A vertical partitioning algorithm for

relational databases, " *in Proc. 3th IEEE Data Eng.*, 1987.

[6] Hoffer, J. A. and D. G. Severance, " The use of cluster analysis in physical database design, *"in Proc. First VLDB,* 1975

[7] Hoffer, J. A., " An integer programming formulation of computer database design problems, " *Inform. Sci.* Vol. 11, pp. 29-48, 1976.

[8] Navathe, S., K. Kamalakav, and M. Y. Ra, " A mixed fragmentation methodology for the initial distributed database design, " *College of Computing Georgia Institute of Technology,* 1992.

[9] Navathe, S., S. Ceri, G. Wiederhold, and J.Dou, "Vertical partitioning algorithms for database design, " *ACM Trans. Database Syst.*, Vol.9, No.4, pp. 680-710, 1984.

[10] Navathe, S. and M. Ra, " Vertical Partitioning for database design: A] graphical algorithm, " *in Proc. ACM SIGMOD Int. Conf. Management Data*, 1989

[11] Ra, M., " A graph-based horizontal partitioning algorithm for distributed of database design, " *Journal of thr Korea Information Science Society,* Vol. 19, No. 1, 1992

[12] Ceri, S., S. Navathe and G. Wiederhold, " Distribution design of logical database schema, " *IEEE Trans. Software Eng.,* Nol. SE-9, No.4, pp.487-503, 1983

[13] Dewan, R. M. and B. Gavish, "Model for the combined logical and physical design of database." *IEEE Transactions on Computers,* Vol.38, No.7, pp.955-967, 1989.

[14] Hammer, M. and B. Niamir, " A heuristic approach to attribute partitioning, " in Proc. *ACM-SIGMOD International Conf,* On Management of data, 1979.

[15] Kusiak, A. And C. H. Cheng, " A branch-and-bound algorithm for solving the grouping thechnology problem," *Annals of Operations Research,*Vol.26, pp.415-431, 1990

[16] Kusiak, A. and W. S. Chow, " An efficient cluster identification algorithm," *IEEE Trans. on Syst., Man and Cybern.,* Vol. SMC-17, No.4, pp.696-699, 1987.