그리디 기반의 연속질의 선택조건의 그룹 필터 순서

김성현[O], 이원석
연세대학교 컴퓨터과학과
{afshkim[O], leewo}@database.yonsei.ac.kr

Greedy based Ordering for Group Filter Processing of Continuous Queries' Selection Predicates

SungHyun Kim[O], WonSuk Lee
Department of Computer Science, Yonsei University

요 약

데이터 스트림 환경에서는 빠르게 무한히 생성되는 데이터를 가능한 빠른 시간에 질의 조건의 충족 여부를 판단하는 것이 시스템의 성능에서 중요한 역할을 한다. 따라서 신속한 판단을 위해 다수의 데이터 스트림 시스템에서 개별 선택 조건을 처리하는 것보다는 속성으로 통합하여 그룹 처리하는 방식을 사용하고 있다. 그룹 필터 처리시에는 그룹 필터의 순서에 따라 큰 처리비용의 차이가 발생하는데 본 논문은 데이터스트림 환경에서 시스템이 처리해야 하는 다수의 연속질의의 선택조건을 그룹 처리할 때 그룹 필터 연산의 처리 비용을 최소화 하기 위한 순서를 결정하는 방법을 제안한다.

1. 서 론

전통적인 DBMS는 데이터를 물리적인 디스크에 저장하고 계속해서 데이터가 누적되는 형태이며 사용자 또는 응용프로그램이 보낸 질의를 DBMS가 대신하여 데이터에 접근하여 그 수행 결과를 요구자에게 보내준다. 그러나 최근에는 이러한 개념의 적용이 곤란한 새로운 형태의 데이터 형태인 데이터 스트림에 관련된 데이터베이스 어플리케이션에 대한 요구가 증가하고 있다. 데이터 스트림은 실시간적이고, 연속적이며, 빠르고, 무한하게 새로운 데이터가 발생한다[1]. 이러한 특징으로 모든 데이터에 대한 일회성 질의(one-time query)가 아니라 질의를 미리 등록해 놓고 데이터가 발생할 때마다 또는 주기적으로 수행 결과를 알려주는 연속 질의(continuous query)의 형태를 가진다[2]. 일회성 질의는 한번 수행되고 시스템에서 제거된다. 그러나 연속질의는 데이터 스트림이 도착할 때마다 계속해서 수행되거나 또는 새로운 윈도우가 시작할 때마다 반복해서 그 질의가 수행된다. 또한 연속질의는 정적인 상태로 저장되기 때문에 많은 연속 질의의 조건을 개별적으로 처리하는 것보다는 연산결과를 공유하는 그룹쿼리 플랜을 생성하거나 또는 그룹 처리방식을 통한 인덱싱으로 처리 시간을 단축한다. 본 논문에서는 연속 질의의 각 조건에 대한 선택률(selectivity)정보를 이용하여 속성별로 그룹 처리할 경우 그룹 필터의 순서 결정 방법을 제안한다. 논문 구성은 다음과 같다. 2장에서는 기존 시스템에서 선택질의를 그룹 처리하는 방식과 필터의 순서에 대한 관련 연구들을 살펴보고 3장에서는 속성별 그룹 처리 시 그룹 필터의 순서를 결정하기 위한 방법을 제안한다. 4장에서는 실험을 통해 성능을 비교하고 5장에서는 결론을 맺는다.

2. 관련연구

NiagraCQ[3]는 XML 문서에 대한 질의 결과를 처리하고 관심에 대상이 되는 웹 문서의 변화를 감시하기 위한 시스템이며 Niagara 프로젝트의 일부분으로 연속질의 처리를 위해 개발되었다. 다수의 질의에 대하여 질의 수행 최적화기가 질의 형식을 분석하여 유사한 표현의 질의를 그룹화하여 계산을 공유한다. 그룹화는 그룹 상수테이블이라는 구조체로 만들어지며 다수의 질의에서 속성과 연산자가 같다면 동일한 상수테이블에 저장되어 선택조건이 데이터와 그룹상수 테이블간의 조인형태로 처리된다.

CACQ[4]는 고정된 쿼리플랜을 수행하지 않고 실행 시 계속해서 연산자의 순서를 동적으로 바꾸는 Eddy[4]에 기반한 시스템이다. CACQ에서는 질의의 선택 조건을 사용된 속성과 연산으로 그룹 필터(Grouped Filter)라는 방식으로 조건을 인덱싱하여 속성과 연산자로 그룹 처리한다. 그룹 필터는 선택 질의에 나타나는 각 데이터 스트림의 속성별로 만들어지며 2개의 해쉬 구조와 2개의 AVL트리 구조로 조건에 사용된 상수가 연산자의 종류에 따라 분리되어 인덱싱된다. 새로운 질의가 등록되면 선택조건에 사용된 속성의 그룹 필터가 있으면 기존의 그룹 필터에 추가되고 없으면 새로운 그룹 필터가 만들어 진다. 데이터 튜플은 그룹 필터를 통해 처리되고 그 결과를 해당 속성이 사용된 모든 질의가 공유한다. 시스템에서 그룹 필터는 하나의 선택 모듈(Selection Module)이 되고 데이터 튜플은 Eddy에 의해

라우팅된다. 다수의 선택모듈이 있을 경우 Eddy는 티켓 라우팅 또는 랜덤의 방식으로 선택한다.

PSoup[6]은 연속질의 뿐만 아니라 임시(ad-hoc)질의의 처리를 지원하기 위해 데이터와 질의를 동기적으로 처리한다. 새로운 데이터는 현재 등록된 과거의 질의에 적용될 수 있고 새로운 질의도 현재 저장되어 있는 과거 데이터에 적용 가능한 시스템이다. 이 시스템에서는 RB-tree라는 데이터 구조를 이용하여 질의의 조건을 인덱싱한다. 트리는 각 속성마다 생성되며 트리의 각 노드는 조건에서 상수를 의미하고 각 노드는 연산자 수 만큼의 배열구조를 가지고 있어 조건의 상수값과 연산자로 질의를 저장한다. 데이터가 처리는 "=" 연산만 사용되었을 경우에는 인덱스 탐색을 통하여 관련된 질의를 찾고 부등호 연산이 사용된 경우에는 순차탐색으로 관련된 모든 질의를 검색한다.

STREAM[1]은 다수의 데이터 스트림이나 저장된 관계 테이블상에서 연속 질의를 처리하기 위한 범용 데이터 스트림 시스템으로 개발되고 있다. 선언적 질의와 동적인 질의 수행계획을 지원하며 적절한 자원 할당과 사용을 통하여 필요에 따라 근사값으로 질의 결과를 대체함으로써 빠른 속도로 생성되는 데이터와 많은 수의 연속질의를 다루도록 설계되었다. 이 시스템은 각 질의에 있는 다수개의 필터 연산의 순서를 정하기 위해 A-Greedy(Adaptive Greedy)라는 알고리즘과 A-Greedy의 변형된 알고리즘을 제시하는 데 실행 시 샘플링을 통하여 필터의 상호 관계를 고려한 조건부 선택률을 구하여 동적으로 순서를 조정한다[7].

3. 속성 그룹 필터의 순서 결정

선택 조건이 속성에 따라 인덱싱되어 있는 속성 그룹 필터가 n 이며 가능한 순서는 n!이고 이중 가장 최선의 순서는 속성 필터를 점검하는 비용이 동일하다고 가정하면 최소의 속성 그룹 필터를 점검하는 것이다. 질의가 하나인 경우에는 다수의 조건이 사용되었다면 무조건 가장 높은 선택률을 가진 속성을 먼저 선택하고 나머지는 상호관계를 고려한 조건부 선택률로 순서를 정하는 것이 최선일 것이다. 하지만 속성별로 그룹처리를 할 경우에는 이와는 접근을 필요로 한다. 질의가 하나인 경우에는 한 조건 "False" 되면 해당 질의는 "False" 되고 해당 튜플이 버려지나 속성별로 질의를 그룹 처리한 경우에는 모든 질의의 조건이 "False" 될 때에만 버려질 수 있다. 예를 들어 한 속성그룹에서 t 개의 질의 중 t-1 개의 질의를 fail 시켜도 나머지 하나의 질의에 대한 조건을 점검할 때까지 계속 진행된다. 그 하나의 질의에 답을 주는 조건이 순서상 마지막에 있다면 False 되는 튜플들은 불필요하게 많은 속성 그룹 필터를 점검하게 된다. 따라서 많은 질의에 연관되어 있는 속성 그룹 필터를 먼저 선택하는 것이 유리하며 한 속성이 얼마나 많은 질의에 연관되었는지를 속성 연관성이라 한다.

정의 1] 속성 연관성
질의에 사용된 n개의 속성을 가진 집합 A = {$A_1, A_2, \cdots, A_n$}라 하면 특정 속성 A_i의 속성 연관성 AA는 다음과 같이 정의한다.

$$AA(A_i) = \frac{A_i\text{가 사용된 질의 수}}{\text{전체 질의 수}} \quad (1<i>n) \tag{1}$$

튜플이 버려지기 위한 최소한의 조건은 모든 질의의 조건을 하나이상 수행해야 된다. 따라서 튜플이 이 조건을 빨리 만족할 수 있도록 순서상 앞으로 배치한다. 속성 연관성이 1 인 속성이 없다면 속성의 조합으로 모든 질의에 연관된 속성 집합을 찾는다. 이 집합에서 원소의 개수는 튜플이 필터 되기 위해 반드시 점검해야 될 속성이므로 적을수록 유리하며 가장 적은 원소를 가진 것을 최소 필터 속성 집합이라 한다.

정의 2] 최소 필터 속성 집합
t 개의 질의가 사용되었다면 질의 집합은 Q={$Q_1, Q_2, \cdots, Q_t$}이며 한 속성 A_i 가 k 개의 질의에 조건으로 사용되었다면 질의 집합 $Q(A_i)$={$Q_1, \cdots, Q_k$}이다. 속성 집합 A 의 부분 집합 중 하나를 S 라 할 때 S 의 원소(속성)가 j 개 일 때 집합 S={$A_1, A_2 \cdots A_j$}(단, S≠{})로 표현한다. S 의 각 원소인 속성의 질의 집합의 합집합이 질의 집합과 같으면 즉 $Q(A_1) \cup Q(A_2) \cup \cdots Q(A_k)$ = Q 이면 S 는 모든 질의의 조건을 점검하는 속성 집합이다. 이러한 S 집합 중에서 원소의 개수(n(S))가 최소인 집합을 최소 속성 필터 집합이라 한다.

최소 필터속성 집합을 찾기 위해서는 조합을 이용한다. 조합 nCr(n 개의 속성에서 r 개를 선택하는 조합)에서 조합의 원소 속성에 연관된 질의의 합집합에서 원소가 질의 전체가 될 때까지 r 을 증가시켜 찾으며 이 때 가능한 속성 집합은 최대 n!/r!(n-r)! 이다. 속성과 질의의 관계를 매트릭스로 보고 질의 Q_j 에서 A_i 속성 조건의 선택률을 $P(Q_j, A_i)$ 표현 한다. 최소 필터 속성 집합이 다수개인 경우에는 각 원소 속성의 선택률 평균을 구한다. 이때 속성의 선택률 평균을 $Aavg(A_i)$라고 하면 각 조건에 대한 속성의 선택률 평균은 다음과 같다. (n:속성 수, t:전체 질의 수)

$$A_{avg}(A_1, \ldots A_i) = \sum_{i=1}^{n} \left(\sum_{j=1}^{t} P(Q_j, A_i) \right) / t \tag{2}$$

if $P(Q_j, A_i)$==null then $P(Q_j, A_i)$=1

위에서 $P(Q_j, A_i)$가 null 이라는 것은 Q_j의 질의에 A_j의 속성에 관련된 질의가 없다는 것이며 이는 곧 모든 튜플이 질의를 만족한다는 의미이다.

예제1] [표1]과 같이 질의와 속성의 관계를 표현한 매트릭스가 있다. 예를 들면 Q_1은 A<3 and C<2라는 조건을 가진 질의이며 ()은 해당 조건의 선택률이다.

1. 최소 속성 필터 집합을 찾는다.
   모든 질의에 관련된 하나의 속성이 없으므로2가지

속성의 조합에서 찾아보면 AB,CD가 된다.

2. 집합의 수가 2이상이면 최소 QP를 선택한다.
   $A_{avg}(A,B)$= 1.35 , $A_{avg}(C,D)$=0.6
   AB를 선택한다.
3. 선택된 속성과 나머지 속성과의 QP를 구하고 최소인 속성을 선택한다
   $A_{avg}(A,B,C)$=1.175 $A_{avg}(A,B,D)$=1.325
   ABC를 선택한다.
4. 모든 순서를 찾을 때 까지 3단계를 반복한다.
   위의 예제에서 결정된 순서는 ABCD이다.

쿼리/속성	A	B	C	D
Q1	A<3(0.3)		C<2(0.2)	
Q2	A>=5(0.5)			D=2(0.1)
Q3		B<4(0.4)	D=2(0.1)	
Q4		B>=8(0.2)		D<4(0.4)

[표 1] 질의와 조건의 매트릭스 표현

4. 실험

본 장에서는 제안한 방법을 실험을 통하여 성능을 비교하였다. 실험환경은 Petium4 CPU 2.66Ghz 메모리 1G,Linux 7.3이며 알고리즘 구현은 C언어로 하였다. 실험에서는 7개의 속성을 사용하였으며 가능한 모든 순서로 데이터가 점검하는 속성그룹의 수를 계산하였다. 그래프에서 WORST인 경우는 가장 많은 속성을 수행할 경우의 순서이고 BEST는 가장 적은 속성을 점검하는 순서이며 Aavg는 3장에서 제안한 방법으로 선택한 순서이다.

[그림 1]에서 실험1은 질의 30개, 조건 147개에 대한 실험 결과이다. 사용된 데이터 셋은 랜덤으로 500만건을 발생시켰다. 각 속성에 대한 각 속성 연관성과 속성에 대한 속성 선택률 평균(A_{avg})는 [표 2]와 같다.

구분	A	B	C	D	E	F	G
AA	0.4	0.5	0.6	0.7	0.8	0.9	1
A_{avg}	0.69	0.61	0.53	0.44	0.34	0.26	0.154

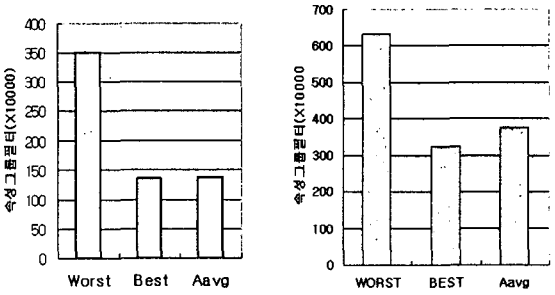
[표 2] 실험1에 사용된 질의의 속성별 특성

실험1에서 최소 필터 속성 집합은 하나이며 F를 원소로 가진다. 따라서 F를 처음으로 시작하며 다음 속성은 F와 나머지 속성의 선택률 평균(A_{avg})를 구하여 최소값을 가지는 속성을 선택한다.

[그림 2] (b)실험 2에서는 1990년 미국 통계자료를 실험 데이터 셋으로 질의는 30개, 조건은 107개를 사용하였다. 질의에 대한 속성의 특성은 [표 3]와 같다.

구분	A	B	C	D	E	F	G
AA	16/30	16/30	10/30	16/30	13/30	13/30	12/30
A_{avg}	0.512	0.585	0.418	0.511	0.598	0.593	0.71

[표 3] 실험2에 사용된 질의의 속성별 특성



(a)실험1 (b)실험2

[그림 1] 성능 비교

최소 필터 속성집합은 {A,C},{C,D} 2개이고 이중 {C,D}의 A_{avg}가 낮아 순서상 처음으로 선택된다. 다른 실험에서도 A_{avg}의 순서가 전반적으로 좋은 성능을 보였다.

5. 결론

데이터 스트림 환경에서는 연속질의의 신속한 처리를 위해 여러가지 질의 처리 최적화 방법이 연구되고 있다. 본 논문에서 선택질의의 그룹 처리시 순서에 따라 시스템의 처리비용이 달라지는 것을 보이고 각 조건의 선택률와 속성 연관성의 정보를 비용함수로 그리디 알고리즘을 적용하여 그룹 필터의 순서를 결정하는 방법을 제안함으로 좋은 순서를 찾을 수 있음을 보였다. 향후 연구 과제로 조건의 선택률이 가용하지 않는 경우에는 추가적인 연구가 요구된다.

참고문헌

[1] Babcock B., Motwani R., Datar M., Babu S., Widom J., *Models and Issues in Data Stream Systems.* In Proc. of the 2002 ACM Sigmod/Sigact Conference

[2] Widom J, Babu S., *Continuous queries over data streams.* SIGMOD Record, 2001: p. 109-120.

[3] J. Chen, D. J.DeWitt, F.Tian and Y.W ang," NiagaraCQ: A Scalable Continuous Query Systemfor Internet Databases," SIGMOD 2000: 379-390.

[4] Avnur R., Hellerstein M. *Eddies: Continuously adaptive query processing.* In ACM SIGMOD, 2000 , Dallas,TX,

[5] Madden S., Shah M., Hellerstein M. & Raman V, *Continuously adaptive continuous queries.* In Proc of SIGMOD Conference, Wisconsin, Madison, June 2002

[6] Franklin M., Chandrasekaran S., *Streaming Queries over Streaming Data.* In 28th VLDB Conference, August 2002

[7] S. Babu, R. Motwani, K. Munagala, I. Nishizawa, and J. Widom. *Adaptive Ordering of Pipelined Stream Filters, SIGMOD, June 2004*