데이터 스트림에서 다중 연속질의의 선택 조건에 대한 실행 순서 결정

윤은원○ 이원석
연세대학교 컴퓨터과학
bigworld77○@database.yonsei.ac.kr, leewo@database.yonsei.ac.kr

Run-time Evaluation of Selection Predicates in Multiple Continuous Queries over Data Streams

Eun Won Yoon○ Won Suk Lee
Department of Computer Science, Yonsei University

요 약

무한히 연속적으로 발생하는 데이터 스트림에서의 연속 질의 처리는 빠른 처리 시간과 적은 메모리 사용량을 요구한다. 이런 제약 사항을 만족하기 위해 연속 질의의 선택 조건절에 사용된 같은 속성들로 그룹화하여 해당 속성들을 처리함으로써 빠르게 질의를 처리할 수 있다. 그리고 더 효율적으로 질의를 처리하기 위해 초기에 일정 기간 동안 데이터 스트림에 대한 통계 정보를 수집한다. 실행 시 통계 정보를 수집하는 이유는 데이터 스트림의 특성을 예측할 수 없기 때문에 데이터 특성에 대한 정보를 수집하고 수집된 정보를 가지고 가장 좋은 질의 처리 순서를 결정함으로 써 전체적인 질의 처리 성능을 향상 시킬 수 있고 실험을 통해 이를 검증한다.

1. 서 론

기존의 데이터베이스 시스템(DBMS)는 물리 저장 공간에 저장된 데이터를 대상으로 사용자나 어플리케이션에서 필요 시 한번 질의를 요청하고 정확한 질의 결과를 가져오는 것이 주요한 목적이었다면 최근의 많은 어플리케이션은 예를 들어 네트워크 감시, 인터넷 상의 고객행위 흐름, 통화기록, 멀티미디어데이터 흐름 그리고, 상품 유통 흐름 등 빠르고 연속적으로 발생하는 데이터 스트림을 대상으로 사용자나 어플리케이션의 요청으로 질의를 수행하는 것이 아니라 새로운 투플이 도착될 때 마다 반복적으로 질의를 수행하여 사용자나 어플리케이션에게 알려줄 것을 요구하는 어플리케이션이 늘고 있다. 이와 같이 기존의 DBMS와는 다른 미들웨어로 데이터 스트림 관리 시스템(DSMS)이 필요하고 DSMS에서 사용하는 질의는 DBMS의 일회성의 질의와는 달리 질의를 미리 등록해서 연속적으로 계속 수행되는 연속질의라고 한다. 그러므로 연속질의는 실시간으로 처리되어야 하며 메모리와 처리 시간에 대한 제약을 수반한다. 이러한 점에서, 다중 연속질의에 대해 개별 질의를 별도로 수행하는 것은 실시간 처리에 있어서 시스템의 부하로 작용할 수 있다.

이 논문은 0000년도 정부(과학기술부)의 재원으로 한국과학재단의 국가지정연구실사업으로 수행된 연구임 (No.M10600000225-06J0000-22510).

```
SELECT *
FROM R
WHERE p1 AND p2 AND ..... AND pk
```

그림 1 일반적인 질의 형태

보다 효율적인 처리를 위해서는 등록된 질의들을 개별적으로 처리하기보다 연속 질의들의 선택조건들의 사용된 공통 속성들의 공유를 통하여 집단적으로 처리하는 것이 유리하다. 이러한 방법은 이미 대부분의DSMS에서 적용하고 있다. 그림 1와 같이 일반적인 질의 형태를 보면 선택 조건절에 p는 데이터 스트림 R의 속성들로 구성되어 있으며, p는 논리곱의 형태로 이루어져 있다. 만약 선택 조건절 p_1 조건을 만족하지 않으면 나머지 $k-1$개 P 조건에 대해서는 더 이상 처리할 필요가 없게 된다.

본 논문에서는 이와 같은 논리곱 형태의 선택 조건들에 대해 연속질의의 집단적인 처리를 위해서 선택 조건절의 속성 그룹의 새로운 구조를 제안한다. 주어진 연속질의 집합에서 적어도 하나의 선택조건에 사용되는 데이터스트림의 속성을 속성 그룹 필터라고 정의하고 속성 그룹

필터는 주어진 다중 연속질의들의 선택조건들 중에서 해당 속성 그룹에 대한 선택조건들을 통합적으로 처리하는 구조체로서 여러 개의 연속질의들 사이에 동일한 속성에 대한 선택조건들이 해당 속성 그룹 필터에 의해 공유된다. 또한 주어진 연속질의들에 대한 하나 이상의 속성 그룹 필터들의 처리순서는 전체적인 질의 처리 성능에 많은 영향을 미칠 수 있다. 왜냐하면, 속성 그룹 필터의 선택률은 서로 간에 차이가 있기 때문이다. 질의최적화의 관점에서 불필요한 연산 수행을 막기 위해서 여러 개의 속성 그룹 필터들 중에서 선택률이 가장 낮은 것을 먼저 수행함으로 써 불필요한 연산 수행을 막을 수 있다. 본 논문에서는 만족되지 않는 스트림 투플의 제거나 연속 질의처리를 가능한 빨리 수행하기 위해서 속성 그룹 필터의 처리순서를 결정하는 방법을 제안한다. 이러한 방법은 질의수행비용을 줄이는데 기여한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 본 논문과 관련된 기존 연구들을 살펴본다. 3장에서는 속성 그룹 필터 처리 방법에 대해 살펴본다. 4장에서는 질의 최적화를 위한 속성 그룹 필터 정렬 방법을 소개한다.5장에서는 제안된 방법의 성능을 실험들을 통해 분석 검증하고, 6장에서 결론을 맺는다.

2. 관련 연구

데이터스트림에 대한 연속 질의처리는 최근 큰 관심을 받고 있고, STREAM[2], NiagaraCQ[3], CACQ[5], 그리고 PSoup[6]을 비롯한 많은 연구들이 제안되었다. 발표된 연구들의 대부분은 다중 연속질의처리를 위한 그룹 기반의 처리기법을 사용한다.

Eddy[4]는 데이터스트림의 대기 큐로부터 새로운 투플을 받아서 라우팅 정책에 따라 선택모듈 중 하나에 보냄으로써 질의를 처리한다. 라우팅 정책에는 랜덤 라우팅과 티켓 라우팅이 있다. 랜덤 라우팅은 새로운 투플을 임의로 라우팅하는 방식이다. 반면, 티켓 라우팅에서는 각각의 선택모듈이 자신의 티켓수를 가지며, 새로운 투플이 특정 선택모듈에서 제거되면 그 선택모듈의 티켓수를 증가시키고 그렇지 않으면 감소시킨다. 결과적으로 많은 티켓을 가진 선택모듈이 많은 투플을 제거한 결과이기 때문에 그 선택모듈로 새로운 투플을 먼저 보내게 된다. 티켓 라우팅의 단점은 매 투플마다 다른 질의 처리순서를 가지는 데이터 스트림의 컨텐츠 기반의 라우팅 방법을 사용해서 빈번한 스케줄 변경이 발생할 수 있다는 점이 질의 처리 성능의 부담으로 작용할 수 있다.

Eddy[4]의 질의처리 구조에 기반을 둔CACQ[5]는 다중의 선택조건들을 동시에 처리하기 위해서 그룹필터라는 기법을 사용한다. 이 기법은 속성기반 그룹화 기법이라는 측면에서 본 논문의 방식과 유사하다. 그룹필터는 선택조건들을 공유함으로써 연산비용을 줄이는 조건색인 연산자로 정의할 수 있다. 연속질의들은 그룹필터를 통해서 수행되며 같은 선택조건들을 가진 질의의 결과는 공유된다. 그러나 본 논문의 방식과는 달리, 범위연산의 경우 조건색인의 특정범위를 순차적으로 검색해야 하기 때문에 질의의 성능이 급속히 저하될 수 있다. STREAM[2]에서는 필터링 순서를 정하기 위해서 조건부 필터 선택률에 기초한 A-Greedy 알고리듬을 사용한다. 실시간으로 수집한 선택률 통계에 따라 적응적으로 필터링 순서를 재 정렬한다. 그러나 단일 질의 최적화에만 적용된다는 면에서 본 논문의 방법과는 다르다.

3. 속성 그룹 필터 처리 방법

데이터 스트림의 한 속성 값 D_i에 대하여 속성 A_i의 모든 조건 $P(A_i)$을 수행한 후 t개의 질의가 가질 수 있는 질의 상태가 q개라면 속성 A_i의 질의 상태 집합 $S(A_i)=\{S_1,S_2,...,S_q\}$이다. 연속질의 선택 조건에 사용된 한 속성의 상수 집합 $C(A_i)$의 상수를 정렬하면 데이터 도메인 범위에서 상수는 점으로 상수와 상수사이는 구간으로 표현된다. 즉 질의에 사용된 속성 A_i의 조건에 k개의 상수가 사용되었다면 모든 질의의 결과는 최대$2k+1$개의 영역으로 나누어진다. 선택 조건에 사용된 상수에 대한 질의 상태$(S_2,S_4\cdots S_{2k})$는 상수와 같은 데이터가 들어 왔을 경우 각각의 질의가 가질 수 있는 상태이며 마찬가지로 구간 값에 대한 질의 상태$(S_3,S_5\cdots S_{2k+1})$는 해당 구간에 속하는 모든 값이 가질 수 있는 질의 상태이다. 따라서 모든 상수와 구간의 대표 값으로 조건을 점검하면 해당 속성에 대한 전체 질의 결과를 실행 전에 알 수 있고 실행 결과를 비트 값1/0으로 표현함으로 써 논리곱 형태의 질의를 빠르게 처리 할 수 있다. 이와 같이 상수 값으로 질의의 결과를 색인한 구조체를 속성 그룹 필터라 한다. 그 구조와 속성 그룹 필터를 이용한 질의 처리 과정은 그림 2와 같다.

속성 그룹 필터는 그림 2와 같이 속성 그룹 필터의 탐색의 키가 되는 상수 값, 결과의 비트열, 그리고 속성 그룹 필터의 순서를 결정하기 위해 각 구간에 대해 투플이 적중된 수를 저장 하고 있어 투플이 발생될 경우 속성 값을 키 값으로 모든 속성 그룹 필터에 대한 결과 상태 비트를 찾아 비트곱 연산을 적용하여 모든 연속 질의의 결과를 판단할 수 있다.

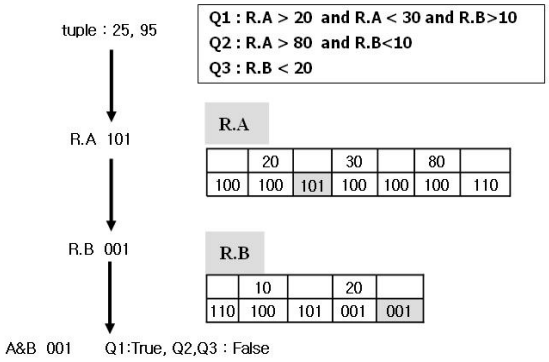

그림 2. 속성 그룹 필터 처리

4. 속성 그룹 필터 순서 결정

연속 질의 처리에 있어서 최적화된 순서를 결정하기 위해서는 순차 선택률을 적용하여 순차 선택률이 가장 낮은 속성 그룹 필터 순으로 처리 순서를 결정함으로 써 불필요한 투플의 처리를 최소화 할 수가 있다. 여기서 순차 선택률을 적용하는 이유는 일반적인 질의 처리 순서의 최적화 방법은 각 속성 그룹들의 선택률을 구하고 선택률이 낮은 순으로 순서를 결정하는데 그 방법은 속성 그룹들 간의 연관성을 고려하지 않게 된다. 따라서 보다 정확한 선택률 정보를 얻기 위해 순차 선택률을 적용한다. 그러나 데이터 스트림의 특성상 발생하는 투플에 대한 정보를 연속 질의의 처리 이 전에는 그 특성을 알 수 가 없다. 따라서 속성 그룹 필터의 방법으로 연속 질의의 처리 순서를 결정하기 위해서는 투플을 처리하는 과정에서 순서를 결정해야만 한다. 따라서 초기에 일정 기간 동안 투플의 통계정보를 수집하고 그 후에 그 정보를 이용하여 속성 그룹 필터의 순서를 동적으로 재결정한다. 일정 기간 동안 모아진 통계정보는 실제 투플을 처리하여 발생된 적중 수를 말하고 정의 1의 방법으로 순서를 결정한다.

정의 1. 순차 선택률

순차 선택률은 등록된 연속 질의 집합 $Q=\{q_1,q_2,..q_k\}$에 표현된 속성들의 집합 $A_p(Q)=\{A_1,A_2,..A_i\}$에 대한 속성의 순서 $\vec{S}=A_1\rightarrow A_2\rightarrow..\rightarrow A_i$의 선택률을 시스템이 시작하는 초기에 일정 기간 λ 동안 발생한 투플에 대해서 식 1을 가지고 계산하는 것을 말한다

$$SA_\lambda(\vec{S})=\frac{P(\vec{S},t_i)}{|Q|*|\lambda|}$$ □

식 1. 순차 선택률 부등식

여기서 $|Q|*|\lambda|$는 λ기간 동안 발생한 투플이 전체 연속 질의를 만족한 수이고 $P(\vec{S},t_i)$는 실제 속성 그룹 필터를 통해 발생한 투플이 해당 연속 질의를 만족한 수가 된다.

예를 들어 그림 1와 같은 상황에서 λ기간 동안 3개의 투플이 발생하였고 A→B를 통과한 투플이 2개일 때 순차 선택률은 2/9가 된다. □

순차 선택률을 통하여 속성 그룹 필터에 대해 반복적으로 적용해서 순서상 순차 선택률이 작은 속성 그룹 필터를 맨 앞으로 결정함으로 써 필요 없는 속성 그룹 필터의 처리를 줄일 수 있다.

그림 3은 속성 그룹 필터의 순서 결정 알고리즘이다. 속성 그룹 필터의 처리 횟수를 줄이는 것은 결과적으로 메모리나 투플의 처리 속도를 향상 시킬 수 있다.

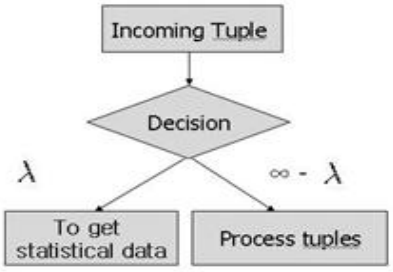

그림 3. 순서 결정 알고리즘

5. 실험 결과

본 장에서는 제안한 방법을 실험을 통하여 성능을 비교하였다. 실험환경은 Petium4 CPU 2.66Ghz 메모리 1G, Linux 7.3이며 알고리즘 구현은 C언어로 하였다. 실험에 사용한 비교함수는 연속 질의를 처리하기위해 투플 당 처리한 속성 그룹 필터의 수를 계산하였다. 그래프에서 WORST Case인 경우는 가장 많은 속성을 수행할 경우의 순서이고 BEST Case는 가장 적은 속성을 점검하는 순서이다. SA는 본 논문에서 제안한 순차 선택률로 결정한 순서이다. 실험에 사용된 연속 질의의 특성은 표1와 같고 데이터 특성은 표 2와 같다.

표 1. 실험 질의 특성

	Q1	Q2
연속질의 수	50	30
속성그룹필터 수	10	7
선택조건 수	204	104

표 2. 실험 데이터 특성

D1	각 속성의 값이 일정한 분포를 가짐
D2	Real Data(A million US Census 1990)

그림 4 같은 경우 Q1의 연속 질의를 D1의 실험 데이터에 적용하여 실험한 경우이다. 이 경우 D1의 데이터 셋의 특징상 유사한 패턴의 데이터들이 많이

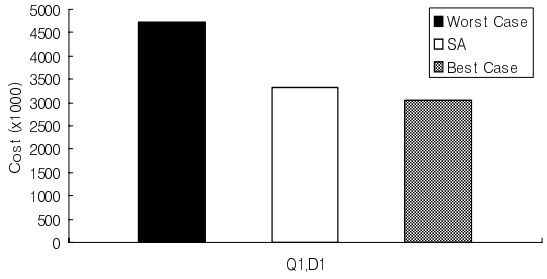

그림 4. 성능 평가

발생하는 경우이다. 최상의 경우와 최악의 경우를 비교해 보았을 때 비교적 최상의 경우에 가까운 성능을 보임을 알 수 있다.

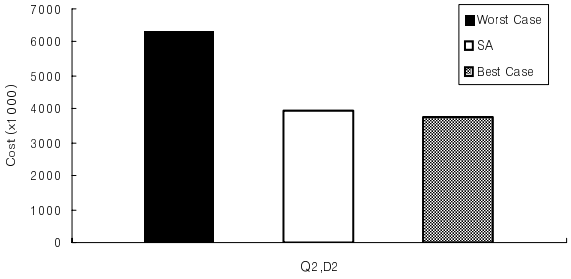

그림 5. 성능 평가

그림 5 와 같은 경우 Q2의 연속 질의를 실제 데이터에 적용하여 실험한 경우이다. 실제 데이터이기 때문에 데이터 셋의 특징을 알 수가 없지만 최상의 경우와 최악의 경우를 비교해 보았을 때 그림 4와 마찬 가지로 최상의 경우에 가까운 성능을 보임을 알 수 있다. 두 실험을 통해 실험 데이터 셋과 실제 데이터 셋에 대해 좋은 성능을 보이는 것을 알 수 있다.

6. 결론

현재 데이터 스트림 환경에서는 연속 질의의 효율적인 처리를 위해 여러가지 질의 처리 방법이 연구되고 있다. 본 논문에서 연속 질의의 선택 조건의 그룹 처리 시 순서에 따라 처리비용이 달라지는 것을 보이고 순차 선택률을 통해 속성 간의 연관성을 고려한 비용함수로 그리디 알고리즘을 적용하여 속성 그룹 필터의 순서를 결정하는 방법을 제안함으로 좋은 순서를 찾을 수 있음을 보였다. 향후 연구 과제로 좀 더 좋은 비용함수에 대해 추가적인 연구와 함께 조인연산이나 집계연산에 대한 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