데이터 스트림에서 다중 조인 질의의 최적화 기법

박연경, 이원석
연세대학교 컴퓨터과학과
e-mail:claire_park@database.yonsei.ac.kr
leewo@database.yonsei.ac.kr

Optimization of Multiple Join Queries over Data Streams

Yon Kyoung Park, Won Suk Lee
Dept. of Computer Science, Yonsei University

요 약

최근 산업발달과 더불어 금융, 의료, 건설 등 다양한 산업분야에서는 대용량의 데이터 들이 실시간에 연속적으로 빠르게 발생되는 경우가 많다. 이런 스트림데이터 형태의 경우 전통적인 DBMS에서 처리하는 방식으로는 모든 데이터를 처리하는 것이 불가능하기 때문에 기존의 방식과 다른 데이터 처리 방식이 요구된다. 본 논문에서는 데이터 스트림에 대한 다중 연속 질의들 사이에서 2개 이상의 스트림을 조인하는 다중 조인 연속 질의를 효율적으로 처리하는 방법을 연구하였다. 다중 조인 연속 질의에 사용되는 조인 조건들 가운데 공통으로 사용된 조인 조건을 공유해 불필요하게 반복되는 질의 수행을 최소화시키고 공통부분을 우선적으로 수행시킴으로써 그 조인 결과의 공유 최대화 및 질의 수행비용의 최소화 할 수 있는 질의 수행 최적화 기법을 제안하고 실험을 통해 제안된 공유 기반의 질의 수행 최적화 기법을 검증하고자 한다.

1. 서론

최근 다양한 산업분야에서 센서 및 네트워크를 활용하는 비즈니스 활동이 활발해 지고 있으며, 물류, 건설 등을 중심으로 활용이 증가하고 있는 추세이다[5]. 또한 주식, 금융, 의료분야 등에서도 다양한 데이터들이 실시간으로 생성되고 있다[10]. 이러한 데이터들은 스트림 형태로 전송되며, 실시간으로 네트워크를 통해 빠르게 모니터링 된다[6].

실시간으로 생성되는 데이터 스트림은 데이터양이 무한하고 가변적이어서 사용자는 새로운 데이터가 들어 올 때 마다 새로운 결과를 요구하기 때문에 한 번 요청된 질의가 계속적으로 실행되어야 한다. 이러한 질의를 일반적으로 연속 질의(continuous query)라고 부른다[1]. 이러한 연속 질의를 전통적인 DBMS에서 이루어지는 한정된 테이블 사이의 연산으로는 처리할 수 없기 때문에 기존의 방식을 데이터 스트림에 적용시키기는 어렵다.[2][8]. 최근 데이터 스트림에서 조인 연산에 윈도우 개념[4]을 이용하여 스트림 형태의 데이터를 한정시켜 조인 연산을 수행한다. 또한 연속 질의 내의 연산들 중에서 가장 비용이 많이 드는 조인 연산의 경우 연산 비용을 절약하기 위한 최적화 과정이 필요하다[9]. 일반적으로 연속 질의의 경우 윈도우와 실행 간격이 함께 주어지게 되면, 데이터 스트림상태에서 다중 연속 조인 질의를 수행하기 위해서는 질의문의 조인 연산의 실행 순서를 먼저 최적화하고, 다음으로는 최소 시간이 소요 되도록 다중 질의 조인을 결정 하게 된다. 본 연구에서는 다중 조인 질의의 효율성을 높이기 위해 질의 조건을 공유함으로써 조연 연산의 중간 결과를 최대로 공유할 수 있는 최적화 알고리즘을 제안하고 실험을 수행하여 성능을 분석하였다.

논문의 구성은 다음과 같다. 2장에서는 스트림데이터 처리를 위한 다중 조인 질의 연구현황을 살펴보고, 3장에서는 조인 조건의 공유 및 윈도우사이즈를 통한 다중 질의 조인의 최적화 알고리즘을 제안한다. 4장에서는 알고리즘의 성능을 평가 및 검증한다. 그리고 5장에서는 결론 및 향후연구에 대해 말하고자 한다.

2. 관련 연구

데이터 스트림 연구는 데이터 처리에 관련된 여러 연산 중 비용 유발 효과가 가장 큰 조인 연산의

* 본 논문은 2007년도 정부(과학기술부)의 재원으로 한국과학재단의 국가지정 연구실사업으로 수행된 연구임 (No.M10600000225-06J0000-22510).

처리를 최적화하기 위한 연구가 활발히 진행되고 있으며, 한정된 자원을 최대로 활용하여 비용효과를 산정하는 연구도 수행되었다[3][9]. 스트림 데이터의 다중 연속 질의 조인에 관한 최근 연구는 윈도우 조인 조건의 공유를 통한 연구[5][10] 및 연속 질의 조건의 공유를 통한 연구가 수행된바 있다[2][6][10]. 그리고 개발시스템의 센서네트워크 활용을 위하여 이진 조인 연산자를 활용해 센서네트워크 조인을 연구하기도 하였는데 이 경우 센서로부터 측정되는 현상과 윈도우 사이즈가 변수로 작용하였다[4].

3. 공유조건을 활용한 최적화 알고리즘

여기서는 다중 연속 질의에서 2개 이상의 조인 수행 문제와 질의에서 공유 가능한 조인 조건과 이진 조인 비용 모델을 정의하고 윈도우 조인에 대해 간략히 설명한다. 이를 바탕으로 다중 조인 질의를 최적화하는 알고리즘을 제안한다.

3.1 스트림 데이터 처리의 문제점

데이터 스트림 시스템에서 가장 부하를 많이 일으키는 것은 조인 연산이다. 만약 다중 연속 질의에서 2개 이상의 조인 연산이 동시에 수행된다면, 시스템의 부하는 배가될 것이다. 따라서 연속 질의 조인의 수를 최소화 시켜야만 한다.

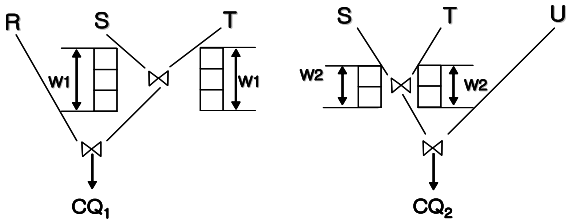
```
CQ1
SELECT *
FROM R [W1 Min], S [W1 Min], T [W1 Min]
WHERE R.a=S.b and S.c=T.d

CQ2
SELECT *
FROM R [W2 Min], S [W2 Min], U [W2 Min]
WHERE S.c=T.d and T.e=U.f
```

(그림 2) 다중 연속 질의의 예

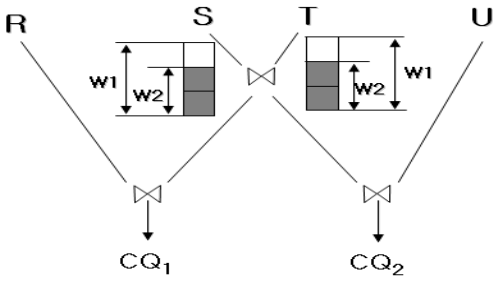
(그림 1)은 CQ_1과 CQ_2는 다중 연속 질의의 한 예이다. 다중 조인 연산을 처리할 때 우선적으로 고려해야 할 점은 조인 연산의 실행 수를 최소화 하는 것이므로 CQ_1과 CQ_2의 질의에 사용 된 조인 조건 가운데 공통으로 사용된 조인 조건을 찾아 다중 연속 질의의 전체 수행 계획 수립 시 조인 연산을 개별적으로 수행 하지 않고 공통된 조인 조건의 일회 실행으로 생성되는 연산의 중간 결과를 재사용 하게 하여 연산의 횟수를 줄인다.

(그림 1)에서 조인 연산의 중간 결과를 공유할 때는 각 질의의 윈도우의 크기를 고려해야 한다. CQ_1과 CQ_2의 입력 스트림의 윈도우 크기가 W_1과 W_2라 한다면, W_1과 W_2가 같을 경우 중첩 윈도우 크기가 같기 때문에 다음 조인에 중간 결과를 내보내는데 문제가 없다. 그러나 CQ_1과 CQ_2의 입력 스트림의 윈도우 크기가 서로 다를 경우 중첩 윈도우의 크기가 다르기 때문에 중간결과를 바로 사용할 수 없음으로 윈도우 크기에 따라 중간 결과를 다음 조인 연산에 적용 가능하도록 조정 해주어야만 하는데, 이러한 문제를 해결하기 위해 튜플 라우팅 연산[7]을 이용해 문제를 해결할 수 있다.

3.2 공유 가능한 조인



(그림 3)각 질의별 수행 계획



(그림 3)중간결과를 공유한 질의 수행 계획

(그림 2)는 (그림 1)의 CQ_1과 CQ_2의 가능한 질의 수행 계획들 중의 하나이고, (그림 3)은 $S \bowtie T$ 조인 연산의 중간결과를 공유하여 질의 연산을 수행하는 경우이다. 중간결과를 공유하여 조인 연산을 수행하기 위해 필요한 공유 가능한 조인을 정의하기 위해 3가지 정의를 설명하면 다음과 같다.

정의 1. 그림 1의 두 개의 CQ_1과 CQ_2 질의에서 공통으로 쓰인 $S \bowtie T$ 조인 조건을 공유 가능한 조인이라 하며 공유 조인이라 말한다.

정의 2. 공유 조인이 n개의 질의에서 사용되어 공유 할 수 있는 횟수를 공유수(Task-num)라 한다.

정의 3. 공유 조인의 공유수가 가장 큰 순서대로 우선적으로 질의 수행을 하도록 구성하는 질의 수행 계획을 네트워크 계획(Network Plan)이라 한다.

만약 각 질의에서 정의한 윈도우 크기가 동일한 경우, 중첩되는 윈도우 크기가 같기 때문에 공유 조인의 중간 결과 전체를 재사용 할 수 있음으로 공유 조인을 우선적으로 수행 하는 질의 계획의 비용이 싸다. 반대로 각 질의의 윈도우 크기가 다를 경우 중첩되는 윈도우 크기가 작아 질 수 있기 때문에 공유 조인의 중간 결과는 전체가 아닌 일부만을 사용할 수 있게 된다. 그러나 일부만을 재사용 한다 하더라도 중간결과를 재사용 하지 않는 것보다는 윈도우 크기가 작다고 하더라도 조인 선택도가 높아 공유할 수 있는 중간 결과 많을 수 있기 때문에 질의 수행비용을 절약할 수 있다. 그래서 각 질의에서 개별적으로 최적의 질의 수행을 하는 것이 수행비용이 항상 작다고 볼 수 없다. 따라서 본 논문에서는 다중 연속 질의에서 공유 조인을 탐색하고 공유 조인

들 중에서 공유수가 많은 조인 연산부터 수행하는 공유 기반의 최적화 알고리즘을 적용한 네트워크 계획을 제시한다.

3.3 조인 비용

조인 조건의 공유를 통한 효과의 평가는 조인비용의 산정을 통해 검증할 수 있으며, 본 연구에서는 네트워크 계획에 대한 조인 비용 계산을 위해 Kang(2003)이 사용한 조인 비용 모델을 참고하였다[3]. 그리고 앞 절에서 언급한 공유 조인에 대한 조인 비용은 공유 횟수에 따라 조인 비용이 감소하나 질의 조인이 공유되더라도 중첩되는 윈도우의 크기가 다를 수 있기 때문에 각 질의에서 정의한 윈도우 크기에 따라 공유 조인에 대한 조인 비용을 각각 계산하여 합산하는 방식을 택하였다. 조인비용 계산식은 식(1)과 같다.

$$C_{R\bowtie S} = \lambda_r(prove(s)+insert(r)+invalidate(r))+\lambda_s(prove(r)+insert(s)+invalidate(s)) \quad (1)$$

조인 연산은 prove, insert, 그리고 invalidate연산이 있으며, invalidate연산은 조인을 수행하면서 이루어지기 때문에 추가적인 비용 없음으로 비용계산에서 제외하였다. prove와 insert연산에 대한 비용 산정 수식은 각각 식(2), (3)과 같고 계산된 값을 식(1)에 대입하면, 전체 조인비용을 계산할 수 있다.

$$prove(r) = \frac{\lambda_s T_s}{|S|} \quad (2)$$

$$insert(r) = 1 \quad (3)$$

각 연산 관련된 용어 정리는 아래 <표 1>과 같다.

<표 1> 용어 정리

$\lambda r, \lambda s$	스트림 R과 S의 입력속도
Tr,Ts	스트림 R과 S의 윈도우 크기
|R|, |S|	윈도우 R과 S의 해시 버켓의 수
R, S	윈도우 R과 S의 튜플의 수
Cost(R,S)	스트림 R과 S의 조인 비용

위 계산식을 정리하면 식(4) 와 같다.

$$C_{R\bowtie S} = \lambda_r(\frac{\lambda_s T_s}{|S|}+1)+\lambda_s(\frac{\lambda_r T_r}{|R|}+1) = \lambda_r\lambda_s(\frac{T_r}{R}+\frac{T_s}{S})+(\lambda_r+\lambda_s) \quad (4)$$

위와 같이 정의된 이진 조인 비용이 공유 조인일 경우 윈도우 크기가 같으면, 다음과 같다.

$$\frac{C_{R\bowtie S}}{Task_{R\bowtie S}} \quad (5)$$

여기서, $Task_{R\bowtie S}$ 는 공유 조인 R과 S의 공유수이다. 만약, 윈도우 크기가 다르다면 조인 비용은 가장 큰 윈도우를 선택해서 조인 비용을 구한다.

3.4 공유기반의 질의 최적화 알고리즘

본 절에서는 지금까지 설명한 것을 바탕으로 공유기반의 최적화 알고리즘을 적용하여 네트워크 계획을 구성하는 다중 연속 질의 최적화 방법에 대하여 설명한다.

조인 테이블은 n개의 다중 연속 질의에서 사용되는 모든 조인 조건을 탐색하고 각 조인 조건들이 공유 가능한 조인인지 알기 위해 공유수를 저장하며 각 조인 조건이 어느 질의에 사용 되었는지에 대한 정보와 윈도우 크기를 저장한다. 가장 공유수가 큰 공유 조인부터 정렬을 한다.

질의 계획 테이블은 조인 테이블의 정보를 이용하여 다중 연속 질의에 대한 네트워크 계획을 구성하는 정보를 저장한다. 조인테이블을 스캔 하면서 공유수가 큰 조인 조건부터 선택하여 질의 계획 테이블에 하나씩 저장하여 네트워크 계획을 구성한다. 본 논문에서 제안하는 알고리즘의 목표는 공유수가 가장 큰 조인부터 수행 될 수 있도록 네트워크 계획을 구성하는 것이기 때문에 조인 테이블에 있는 모든 조인들이 수행 될 때까지 반복 수행 하면 된다. 본 논문에서 제안하는 알고리즘은 (그림 3)과 같다.

```
Make_Network_Plan(Join_Table)
  Join_Table: 다중 연속질의에서 사용된 모든 조인조건들이 공유수가 큰 공유조인부터 저장 되어있는 테이블
  Network_Table: 조인 테이블에서 선택된 조인 정보를 저장하는 테이블
While(Join_Table[i].Flag!=0)//조인 테이블 탐색
  If Network_Table is NULL
    Join_Table의 첫번째 조인을 선택해서 Network_Table에 저장
  else
    Join_Table에서 Network_Table로 선택된 조인조건을 제외한 조인조건들 중에서 공유수가 가장 큰 조건을 선택
    if 이미 Network_Table에 선택된 조인 조건들 중에서 질의 정보가 Join_Table에서 선택된 조인조건의 질의 정보가 부분 집합이라면
      그 조인조건을 선택하지 않고 다음 조인조건으로 넘어간다.
    else
      Join_Table의 조인조건을 선택하여 Network_Table에 저장
```

(그림 3)공유기반의 질의 최적화 알고리즘

이 알고리즘에서 공유수가 크다고 무조건 네트워크 계획에서 먼저 수행이 되는 것이 아니라 이미 네트워크 테이블에 저장되어 있는 다른 공유조인의 질의 정보를 보고 그 질의 정보가 현재 선택하려고 하는 공유조인의 질의 정보와 부분집합을 이룬다면 그 조인은 기존에 네트워크 테이블에 선택된 조인 조건의 중간결과를 재사용할 수 있는 조인 조건이므로 동시에 수행 되지 않도록 한다. 예를 들어, $R\bowtie S$ 는 공유수가 3이고 질의 CQ_1, CQ_2, CQ_3에서 사용되고 $S\bowtie T$ 는 공유수가 2이고 질의 CQ_1, CQ_2에서 사용된다면 이미 네트워크 테이블에 $R\bowtie S$ 가 저장되어 있다면 $S\bowtie T$ 는 $R\bowtie S$ 의 중간결과를 사용해야 하기 때문에 공유수가 나머지 조인조건들 보다 크더라도 현재 네트워크 테이블에 저장할 수 있는 조인 대상에서 제외 되는 것이다.

4. 실험 및 성능평가

본 절에서는 공유기반의 최적화 알고리즘을 적용하여 구성한 네트워크 계획이 다중 연속 질의내의

조인 연산의 비용을 얼마나 효과적으로 줄여주는 지에 대한 실험을 하였다. 실험에 사용한 데이터 셋은 <표 2>와 같다.

<표 2> 실험에 사용한 데이터 셋

질의	3개의 조인 조건이 포함된 질의
질의수	10, 20, 30개
윈도우크기	10, 20, 30, 40, 50에서 랜덤하게 사용
입력속도	0~200 tuples/sec에서 랜덤하게 생성
조인선택도	0.1~0.01의 값을 랜덤하게 생성

다중연속질의 내에 사용된 조인 조건의 공유정도에 따른 성능평가를 하기 위해 공유률(μ)에 따른 질의셋을 구성하였다. 공유률은 식(6)과 같으며, τ와 ϕ는 각각 공유조인의 공유 및 질의에 사용된 조인 조건이다.

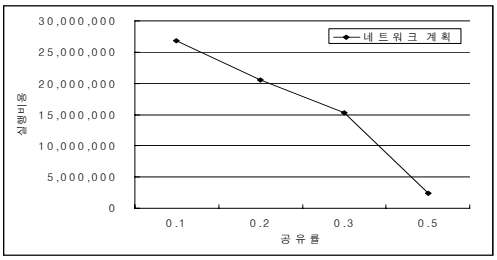
$$\mu = \frac{\sum \tau}{\sum \phi} \qquad (6)$$

실험에 사용된 질의셋은 <표 3>과 같다.

<표 3> 실험에 사용한 질의셋

질의셋	공유율
질의셋 1	0.1
질의셋 2	0.2
질의셋 3	0.3
질의셋 4	0.5

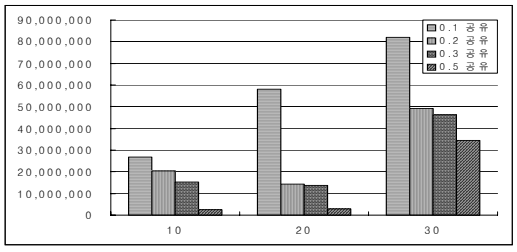
(그림 4)에서는 질의 수는 10개로 고정시키고 질의셋1, 2, 3, 4를 사용하여 공유율에 따른 네트워크 계획의 실행 비용의 변화를 평가하였다.



(그림 4) 공유률에 따른 실행 비용 비교

(그림 4)의 결과에서 알 수 있듯이 질의 내의 공유 조인의 공유률이 높을수록 공유 조인의 중간결과의 재사용이 많기 때문에 실행 비용이 감소됨을 알 수 있다.

(그림 5)에서는 질의 수를 10, 20, 30개로 하여 질의 수가 증가함에 따라 네트워크 계획의 실행 비용의 변화를 평가하였다.



(그림 5) 질의 수에 따른 조인 비용 비교

질의 수가 많아지더라도 질의 내의 공유 조인의 공유율이 높을수록 실행 비용이 감소됨을 알 수 있다.

5.결론 및 향후 연구

데이터 스트림 시스템에서 가장 비용유발이 큰 조인 연산의 수행을 최소화하기 위해 다중 연속 질의 내에 공통으로 사용되는 공유 조인을 탐색하고 그 공유 조인의 중간결과를 공유할 수 있는 전체 질의 계획인 네트워크 계획을 구성하는 질의 수행 계획을 제시하였다. 또한 실험을 통해 공유조인의 중간결과를 재사용 하는 것이 조인 비용의 감소에 큰 영향이 있음을 알 수 있었다.

향후, 다중 연속 질의 연구에서는 2개 이상의 조인 연산에 대한 질의 수행 계획 구성을 위해 공유 조인의 공유수 이외에 스트림의 입력속도, 조인 선택도 등에 따른 조인비용의 변화를 고려한 질의 수행 계획의 구성이 연구 되어야 할 것이다.

참고문헌

[1] Babu S. and Widom J., Continuous Queries Over Data Streams, ACM SIGMOD Record archive volume 30, Issue 3, 2001, 109-120.

[2] B. Babcock, Babu S., M. Datar, R. Motwani, and Widom J., Models and Issue in Data Stream Systems, In Proc. of PODS, 2002.

[3] J. Kang, J. F. Naughton, and S. Viglas, Evaluating window joins over unbounded streams, In Proc. of the 2003 Intl. Conf. on Data Engineering, Mar. 2003.

[4] Lukasz Golab M. Tamer Ozsu, Processing Sliding Window Multi-Joins in Continuous Queries over Data Streams, Proc. VLDB, 2003

[5] Moustafa A. Hammad, Michael J. Franklin, Walid G. Aref, Ahmed K. Elmagarmid, Scheduling for shared window joins over data streams, Proc. VLDB, 2003.

[6] M. H. Ali, W. G. Aref, R. Bose, A. K. Elmagarmid, A. Helal, I. Kamel, and M. F. Mokbel. NILE-PDT:A phenomenon detection and tracking framework for data stream management systems. In VLDB, p1295 - 1298, 2005

[7] Song Wang, Elke Rundensteiner and Ganguly S., Bhatnagar S., StateSlice:New Paradigm of Multi-query Optimization of Window-based Stream Quaries, Proc. VLDB, 2006.

[8] The STREAM groups STREAM : The Stanford Stream Data Manager(short overview paper), IEEE Data Engineering Bulletin, March 2003.

[9] TIMOS K.SELLIS, Multiple-Query Optimization, ACM Transactions on Database Systems, 1988.

[10] Yali Zhu, Elke A Rundensteiner and George T. Heineman, Dynamic Plan Migration for Continuous Queries Over Data Streams, ACM, SIGMOD June 13-18, 2004.