데이터 웨어하우스에서 참조 무결성 제약 조건을 이용한 병렬 뷰 일관성 관리 기법의 성능 평가

이병숙[0] 김진호
강원대학교 전자계산학과
bsdream@bcline.com jhkim@cc.kangwon.ac.kr

Performance Evaluation On Parallel View Consistency Maintenance Using Referential Integrity Constraints in Data Warehouse Environment

Byoung-Suk Lee[0] and Jin-Ho Kim
Dept. of Computer Science, Kangwon National University

요 약

데이터 웨어하우스는 효과적인 질의와 분석을 위해 물리적으로 여러 사이트에 분산된 소스 데이터로부터 통합된 정보를 추출하여 저장한 데이터 저장소로써, 실체 뷰의 집합으로 구성된다. 따라서 데이터 소스에 변경 사항이 발생하면 데이터 웨어하우스와 일관성을 유지하기 위해 실체 뷰에도 변경 사항을 반영하는 뷰 관리가 필요하다. 동시에 변경되는 여러 데이터 소스와 뷰의 상태 사이에 일관성을 보장하기 위해서는 각 소스의 변경 사항을 순서대로 뷰에 반영해야 한다. 이때 각 소스의 변경 사항을 뷰 정의와 관련된 다른 소스들과 조인을 수행해야 하는 등 뷰 갱신을 위해 많은 비용이 소요된다. PSWEEP/RI기법은 이러한 뷰 갱신 비용을 줄이는 방법으로 뷰의 일관성을 보장하기 위해 수행해야 하는 서브질의를 참조 무결성 제약조건의 특성을 이용하여 병렬로 처리하는 방법이다. 본 논문에서는 PSWEEP/RI기법의 성능을 평가하기 위하여 이 방법의 비용 모델을 분석적으로 제시하였으며, 이 모델을 기반으로 다른 기존의 방법(SWEEP)과 성능을 비교 분석하여, PSWEEP/RI 기법이 다른 기존의 방법(SWEEP)보다 여러 소스 릴레이션의 조인으로 구성된 실체 뷰를 갱신하는 시간을 크게 단축하여 효율적으로 뷰를 관리하며, 소스의 증가에 따른 뷰 갱신 시간의 증가를 줄일 수 있음을 보였다.

1. 서 론[1)]

데이터 웨어하우스는 물리적으로 여러 사이트에 분산된 데이터 소스로부터 추출한 온라인 분석 정보를 유지하는 실체 뷰의 집합으로 구성된다. 그러므로 사용자들은 분석과 질의에 DW의 실체뷰를 이용하여 보다 빠른 분석 결과를 얻을 수 있다[1]. DW에 존재하는 데이터 소스들은 지역적으로 떨어져 있으며 서로 독립적으로 수정될 수 있기 때문에, 동시에 여러 소스에 변경사항이 발생하는 상황에서 DW의 뷰를 일관되게 관리할 수 있는 점진적 뷰 관리 기법이 필요하다[2]. 기존의 데이터웨어하우스와 뷰의 일관성을 유지하는 뷰 관리 기법으로 ECA, Strobe, SWEEP, PVM등의 알고리즘들이 소개되었다[1][3][4][5].

ECA 알고리즘은 단일 사이트의 소스 환경에서, Strobe와 SWEEP은 분산 소스 환경에서 일관성을 만족하는 뷰 관리 기법들을 제시하였다[1][3]. SWEEP은 순차적으로 변경사항들을 뷰에 반영하므로 소스의 수가 증가함에 따라 뷰 갱신비용이 비례적으로 증가하게 된다[1]. 따라서 대규모의 DW에 이용되기에 적절하지 못하다. 이러한 뷰 갱신 비용을 줄이기 위해 변경사항들을 병렬로 처리하는 방법이 [4]에서 제시되었다. [4]에서 제시된 병렬처리 알고리즘 PVM은 SWEEP과 동일한 방법으로 변경 사항을 뷰에 반영하는 쓰레드를 여러 개 만들어 병렬로 처리하는 방법이다. 여러 변경사항들을 병렬로 처리함으로써 처리비용을 줄일 수 있지만, 하나의 변경사항에 대한 뷰 갱신값을 얻기 위한 조인들은 여전히 SWEEP과 동일하게 순차적으로 처리하고 있다[4]. 따라서 뷰 갱신 값을 얻기 위해 수행하는 조인 연산들을 병렬로 처리하는 기법인 PSWEEP/RI (Parallel SWEEP with Referential Integrity)이 제시되었다. PSWEEP/RI은 변경 소스와 다른 소스들간의 조인을 병렬화 하기위해 조인관계에 있는 소스 릴레이션들간의 참조 무결성을 이용한다. 즉, PSWEEP/RI는 소스 릴레이션들간의 참조 무결성 관계를 설정한 후, 다른 릴레이션을 참조하는 릴레이션에서 변경사항이 발생할 경우 뷰갱신 값을 계산하는 조인 처리시 참조하는 릴레이션의 개수만큼 병렬로 조인을 수행한다. 또한 참조되는 릴레이션의 변경사항은 뷰에 영향을 주지 않으므로 이 경우는 조인 계산을 하지 않고 필터링한다. 본 논문에서는 PSWEEP/RI기법의 성능을 평가하기 위하여 이 방법의 비용 모델을 분석적으로 제시하였으며, 이 모델을 기반으로 다른 기존의 방법(SWEEP)과 성능을 비교 분석하여, PSWEEP/RI기법이 다른 기존의 방법(SWEEP)보다 뷰의 일관성을 관리하기 위해 요구되는 갱신비용을 크게 감소시키며, 소스가 증가하더라도 조인계산이 병렬처리 되므로 갱신비용이 크게 증가하지 않도록 효율적으로 관리할 수 있음을 실증하였다.

2. 관련연구

실체뷰 관리에 있어 중요한 사항은 데이터 소스와의 일관성(Consistency)유지와 최적의 뷰 관리 비용 문제이다[1][3][5]. 뷰 관리시 데이터 웨어하우스는 소스 데이터에서 발생한 변경사항을 관련된 다른 소스 사이트의 정보와 통합하여 실체 뷰에 반영하게 되는데, 이때 다른 사이트들도 독립적으로 동시에 변경될 수 있기 때문에 각 소스의 변경사항과 일관된 정보들을 다른 사이트로부터 추출하여 뷰를 일관성 있는 상태로 수정/유지하는 것은 상당히 어렵다. 그러므로 동시 변경사항(Concurrent Update)이 발생하는 상황에서 DW의 뷰를 일관되게 관리할 수 있는 점진적 뷰 관리 기법이 필요하다[2]. 이러한 문제점을

1) 이 논문은 첨단정보기술연구센터(AITrc)를 통하여 한국과학재단의 지원을 받았음.

해결하기 위한 접근법으로 ECA, Strobe, SWEEP, PVM등의 알고리즘들이 개발되었다. Strobe와 SWEEP 알고리즘은 분산 소스 환경에서 일관성을 만족하는 뷰 관리 기법들을 제시하였다. SWEEP에서는 일관성을 유지하기위해 발생한 모든 변경사항들을 DW에 도착한 순서대로 UMQ(Update Message Queue)에 저장하여 순차적으로 처리된다. 또한 각 변경사항에 대한 뷰 갱신 값을 계산하기 위해 다른 소스 사이트와 조인하는 서브질의들을 순차적으로 처리한다. 따라서 소스의 수가 많거나 변경사항들이 증가할 경우 처리비용이 상대적으로 증가한다는 단점이 있다. 이러한 단점을 보완하기 위해 변경사항들을 병렬처리하는 방법이 [4]에서 제시되었다. [4]에서 제시된 병렬처리 알고리즘인 PVM은 뷰 갱신 쓰레드를 여러 개 두어 SWEEP과 동일한 방법으로 여러 개의 변경사항들을 동시에 처리하는 개념이다. PVM은 변경사항이 병렬 처리되므로, 병렬 프로세스의 수만큼 변경사항에 대한 뷰갱신 시간은 단축된다. 그러나 이 방법에서도 각 변경사항을 반영하기 위해 필요한 여러 개의 조인 연산등의 서브질의들을 SWEEP에서와 같이 순차적으로 수행하고 있어 이에 대한 수행 시간은 여전히 동일하게 소요된다. 이러한 문제를 해결하기 위해 PSWEEP/RI 기법이 제시되었다. 이 기법은 참조 무결성 제약조건의 특성을 이용하여 병렬로 이들 서브질의들을 처리하는 방법으로 각 변경사항에 대한 뷰 갱신을 효율적으로 수행할 수 있는 방법을 제시하였다. 본 논문에서는 기존 방법에 대한 성능의 비교 연구를 위해 PSWEEP/RI 기법의 비용 모델을 분석적으로 제시하고, 이를 통해 PSWEEP/RI 기법의 성능 평가 및 기존 방법과 비교/분석 하고자 한다.

3. PSWEEP/RI 기법

PSWEEP/RI는 서브질의의 소스와의 조인 연산을 병렬화하기 위해 조인관계에 있는 소스 릴레이션들간의 참조 무결성 제약 조건을 이용한다. 조인되는 릴레이션들간에 참조 무결성 제약 조건을 만족하는 경우, 참조 릴레이션의 변경사항은 여러 피참조 릴레이션들과의 조인을 병렬로 처리할 수 있다. 또한 피참조의 삽입(또는 삭제)의 변경사항은 참조 무결성 제약 조건 때문에 이를 참조하는 투플들이 다른 참조 릴레이션에 존재할 수 없으므로 이들과의 조인을 아예 수행할 필요가 없다. 따라서 본 논문에서 제시하는 PSWEEP/RI는 소스 데이터인 릴레이션들간의 참조 무결성 제약 조건(Referential Integrity Constraints : RI)을 설정한 후, 변경사항이 발생하는 릴레이션의 참조관계에 따라 생성된 서브질의의 조인 연산들을 병렬로 처리하거나 필터링시키는 방법을 제시한다.

예를 들어, V = $R_1 \bowtie R_2 \bowtie R_3 \bowtie R_4 \bowtie R_5 \bowtie R_6$인 데이터 웨어하우스 뷰가 있을 경우, 그림 1과 같이 뷰에서 이용된 각 소스의 기본 릴레이션간에 참조 무결성 제약 조건을 설정한 참조 무결성 그래프를 생성한다. 만약 여러 개의 릴레이션을 참조하는 R_2 릴레이션에서 변경사항($\triangle R_2$)이 발생 한다면, $\triangle V = R_1 \bowtie \triangle R_2 \bowtie R_3 \bowtie R_4 \bowtie R_5 \bowtie R_6$를 계산하기 위해 R_1, R_2, R_3와 $\triangle R_2$와의 조인을 다음과 같이 동시에 병렬 처리 할 수 있다. 또한 R_4와R_5 및 R_6이 참조 관계이므로 이들 역시 병렬처리 될 수 있다. 마찬가지로 변경사항이 발생하는 릴레이션이 다중 피참조 관계인 경우는 참조 무결성 제약조건에 의해 삽입이나 삭제시 뷰 테이블에 영향을 주지 않는다. 그러므로 피참조 릴레이션의 변경사항은 필터링 된다. 따라서 PSWEEP/RI 기법은 병렬 조인을 이용하여 뷰 갱신비용을 단축할 뿐 아니라, 불필요한 질의 처리를 줄일 수 있으므로 SWEEP의 경우 보다 매우 효율적이다.

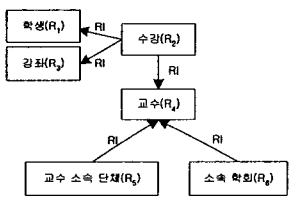

그림 1. 참조 무결성 그래프

4. PSWEEP/RI기법의 비용 모델

이 장에서는 본 논문에서 제시한 [그림 1]의 예제를 처리하는데 소요되는 PSWEEP/RI기법과 SWEEP 기법의 성능을 비교.평가하기 위한 비용 모델을 제시한다. 이 비용 모델은 조인 처리비용과 전송비용으로 구성된다.

4.1 Parameter

비용모델에 사용되는 매개 변수와 식은 다음과 같다.

Ccom : 통신 속도
n_i : table R_i의 tuple 수
w_i : table R_i의 tuple의 크기
nd_i : Delta table R_i의 tuple수
B : block의 크기
k : 일반적인 경우의 깊이 탐색 비용
br : 통신 상수 값

4.2 PSWEEP/RI의 비용 식

[그림 1]의 예제를 처리하기 위한 PSWEEP/RI의 비용식은 다음과 같다. (지면 관계상 SWEEP의 비용 식은 생략한다. 또한 조인 비용은 인덱스 조인과 중첩 루프 조인 모두에 대해 성능 평가를 하였지만 역시 지면관계상 인덱스 조인의 경우만 소개한다.) 여기서, $\triangle R_2$는 참조하는 릴레이션에서 발생한 변경사항이며, DW는 데이터 웨어하우스를 말한다.

① $\triangle R_2 \rightarrow$DW 통신 비용 = $\frac{nd_2 * w_2 * 8}{Ccom}$ [식 4.2.1]

② $\triangle R_{2(DW)} \rightarrow R_1$ 통신 비용 = $\frac{nd_2 * w_2 * 8}{Ccom}$ [식 4.2.2]

② $\triangle R_{2(DW)} \rightarrow R_3$ 통신비용 = $\frac{nd_2 * w_2 * 8}{Ccom}$ [식 4.2.3]

② $\triangle R_{2(DW)} \rightarrow R_4$ 통신비용 = $\frac{nd_2 * w_2 * 8}{Ccom}$ [식 4.2.4]

③ $\triangle R_2 \bowtie R_1$: X 조인 비용 = $nd_2 * \lceil \log_k n_1 + 1 \rceil * br$ [식 4.2.5]

③ $\triangle R_2 \bowtie R_3$: Y 조인 비용 = $nd_2 * \lceil \log_k n_3 + 1 \rceil * br$ [식 4.2.6]

③ $\triangle R_2 \bowtie R_4$: Z 조인 비용 = $nd_2 * \lceil \log_k n_4 + 1 \rceil * br$ [식 4.2.7]

④ X→DW 통신비용= $\frac{nd_2 * (w_2 + w_1) * 8}{Ccom}$ [식 4.2.8]

④ Y→DW 통신비용= $\frac{nd_2 * (w_2 + w_3) * 8}{Ccom}$ [식 4.2.9]

④ Z→DW 통신비용= $\frac{nd_2 * (w_2 + w_4) * 8}{Ccom}$ [식 4.2.10]

⑤ 최대값 선택 비용 :
병렬로 수행되는 각 참조되는 릴레이션 (R_1, R_3, R_4)과의 조인 비용(즉,②+③+④)중 최대값을 선택 [식 4.2.11]

⑥ X $\bowtie$ Y : M 조인 비용 = $\lceil \frac{nd_2*(w_2+w_1)}{B} \rceil * \lceil \frac{nd_2*(w_2+w_3)}{B} \rceil * br$ [식 4.2.12]

⑦ M $\bowtie$ Z : N 조인 비용 = $\lceil \frac{nd_2*(w_2+w_1+w_3)}{B} \rceil * \lceil \frac{nd_2*(w_2+w_4)}{B} \rceil * br$ [식 4.2.13]

⑧ $N_{(DW)} \rightarrow R_5$: 통신비용 = $\frac{nd_2*(w_2+w_1+w_3+w_4)*8}{Ccom}$ [식 4.2.14]

⑧ $N_{(DW)} \rightarrow R_6$: 통신비용 =

$$\frac{nd_2*(w_2+w_1+w_3+w_4)*8}{Ccom}$$ [식 4.2.15]

⑨ N ⋈ R_5 : I 조인 비용 =

$$nd_2*(\lceil \log_k n_5 \rceil + \frac{n_5}{n_4})*br$$ [식 4.2.16]

⑨ N ⋈ R_6 : J 조인 비용 =

$$nd_2*(\lceil \log_k n_6 \rceil + \frac{n_6}{n_4})*br$$ [식 4.2.17]

⑩ I → DW 통신비용 =

$$\frac{nd_2*\frac{n_5}{n_4}*(w_2+w_1+w_3+w_4+w_5)*8}{Ccom}$$ [식 4.2.18]

⑩ J → DW 통신비용 =

$$\frac{nd_2*\frac{n_6}{n_4}*(w_2+w_1+w_3+w_4+w_6)*8}{Ccom}$$ [식 4.2.19]

⑪ 최대값 선택 비용
[그림]에서 피참조 릴레이션의 병렬 조인 (⑧+⑨+⑩)의 결과중 최대 값 선택 [식 4.2.20]

⑫ I ⋈ J 조인 비용 = [식 4.2.21]

$$\lceil \frac{nd_2*\frac{n_5}{n_4}*(w_2+w_1+w_3+w_4+w_5)}{B} \rceil * \lceil \frac{nd_2*\frac{n_6}{n_4}*(w_2+w_1+w_3+w_4+w_6)}{B} \rceil *br$$

전체 비용은 [식 4.2.1] + [식 4.2.11] + [식 4.2.12]+[식 4.2.13] +[식 4.2.20] +[식 4.2.21]이다.

5. PSWEEP/RI의 성능 평가

실험은 통신 속도, 릴레이션의 크기, 변경사항 투플 수를 증가시키면서 PSWEEP/RI와 SWEEP방법에서 각각 소요되는 비용을 계산한다. 기본 릴레이션의 투플수는 n1=300,000, n2=1500,000, n3=100,000, n4=50,000, n5=50,000이다. 통신 속도는 100K에서 100M까지 증가시키면서 소요 비용을 특정하며, 기본 릴레이션의 크기는 300배까지 증가 시켜 조인 비용의 변화를 측정한다.

5.1 통신 속도의 변화에 따른 성능 분석

그림 2는 통신 속도의 변화에 따른 조인 비용에 대한 성능 평가 결과를 보여 주고 있다.

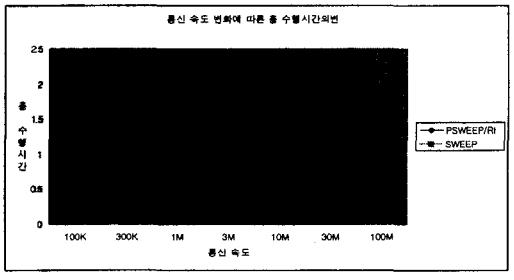

그림 2. 통신 속도 변화에 따른 총 수행 시간의 변화

통신 속도가 높을 때는 SWEEP과 큰 차이가 없으나 통신 속도가 낮을 때는 PSWEEP/RI방법의 성능이 훨씬 우수함을 확인 할 수 있다.

5.2 릴레이션 크기의 변화에 따른 성능 분석

릴레이션 크기의 변화를 통한 성능 분석에서는 변경 투플 수를 1개로 고정시키고, 통신속도를 10M로 고정시킨 후 각각의 릴레이션 크기를 300배까지 동일하게 증가시키면서 성능을 비교한다.

그림 3은 릴레이션 크기 증가에 따른 총 수행 비용을 보여 주고 있다. 전반적으로 PSWEEP/RI 기법이 릴레이션의 크기 증가에 영향을 덜 받음을 알 수 있다.

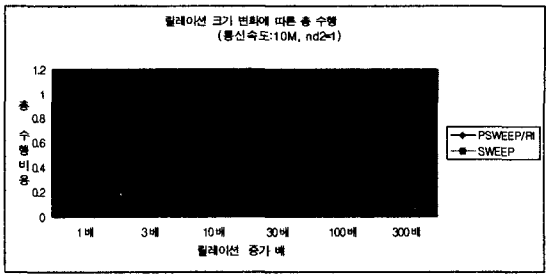

그림 3. 릴레이션 크기 변화에 따른 총 수행시간의 변화

6. 결론

일관성을 보장하는 대표적인 뷰 관리 기법인 SWEEP의 높은 뷰 갱신 시간을 단축하기 위해, 이 논문에서는 다른 소스와의 조인을 병렬로 수행하는 PSWEEP/RI의 처리 기법 연구와 성능 평가를 제시하였다. 이 방법에서 서브질의의 조인 연산을 병렬화 하기 위해 소스 릴레이션들간의 참조 무결성 제약조건의 특성을 이용하였다. 이 방법의 성능을 평가하기 위해 통신 속도, 릴레이션의 크기, 변경사항 투플의 크기등을 증가시켜 가며 기존 SWEEP의 방법과 비교.분석하였다. 성능 평가 결과 뷰 갱신 시간을 크게 단축하여 SWEEP의 기법보다 우수함을 보였다. 따라서 본 기법은 기존 방법인 SWEEP에서 요구되는 뷰 갱신시간을 크게 단축하고, 소스의 증가에 따른 뷰 갱신시간의 증가를 가져오는 SWEEP의 단점을 개선하였다.

앞으로 PSWEEP/RI에서 참조 무결성 관계를 보다 효율적으로 관리할 수 있는 알고리즘 개발이 향후 연구 과제로 남아 있다.

7. 참고문헌

[1] D. Quess, A. Gupta, I. S. Mumick, and J. Widom, "Making Views Self-Maintainable for Data Warehousing", Proc. of Conf. on Parallel and Database Information Systems, pp.158-169, 1996.

[2] K. A. Ross, D. Stivastava, and S. Sudarshan, "Materialized View Maintenance and Integrity Constraint Checking: Trading Space for Time,"Proc. of ACM SIGMOD Conf, pp.447-458, 1996.

[3] D. Agrawal, A. EL Abbadi, A. Singh, T. Yurek, "Efficient View Maintenance at Data Warehouses", In Proceedings of SIGMOD Conference, pp.417-427, 1997.

[4] Xin Zhang, Lingli Ding, Elke A. Rundensteiner, " Parallel Multi-Source View Maintenance", Submitted for publication,2002

[5] Yue Zhuge, Hector Garcia-Molina, Janet L. Wiener, "The Strobe Algorithms for Multi-Source Warehouse Consistency", In PDIS pp.146-157, 1996.

[6] Wooley lee, Il-Yeol Sont, "Efficient Maintenance of Materialized Views for data Warehouse Using Differential Files", Submitted for publication,2002.