관계형 데이터베이스에서 테이블 연산시 효율 향상을 위한 성능 평가 방법

신명주° 김용성
전북대학교 대학원 컴퓨터정보학과
silk@moak.chonbuk.ac.kr, yskim@moak.chonbuk.ac.kr

A Performance Evaluation Method For Efficiency Table Operation in A Relational Database

Myeong-Ju Shin° Yong-Sung Kim
Dept. of Computer Information, Chonbuk National University

요 약

기존의 데이타베이스 시스템에서 주요 성능문제는 대부분 응용프로그램의 SQL문에서 발생한다. 특히 다중 테이블을 이용한 SQL문에서 필수적으로 사용하는 JOIN 연산이 단순 질의 연산에 비해 많은 성능 문제점을 갖고 있다. 따라서 본 연구에서는 관계형 데이타베이스시스템에서 JOIN연산의 효율성을 높이기 위한 방안을 설정하고 이를 적용해 본다.

1. 서 론

오늘날 데이터베이스(database : DB) 시스템의 성능관리는 응용 시스템의 성능을 좌우하는 중요한 요소가 되어가고 있다. 복잡하고 다양한 사용자의 요구에 따라 자료의 양은 방대하고, 방대한 자료에서 정보를 신속하게 제공하기에는 시스템 관리에 많은 노력이 필요로 한다. 시스템의 성능에 영향을 미치는 요소는 대단히 많지만, 대부분 애플리케이션(application) 성능개선으로 효과적인 시스템을 유지할 수 있다.

데이터베이스 시스템의 성능 문제는 약 80%가 응용프로그램의 SQL문에서 발생한다[1]. 따라서 현재 많이 사용하고 있는 관계형 데이터베이스에서 SQL문 작성시 필수적으로 사용되는 조인(JOIN)의 효율을 알아보고, 개선 하는 것이 데이터베이스의 성능 향상에 매우 중요하다.

조인연산은 사용자가 원하는 데이터가 여러 테이블에 있는 경우 특정 컬럼을 서로 연결하여 데이터를 가져오는 방법이다. 따라서 조인연산시 최적의 경로를 찾는 것은 매우 중요하며 SQL을 잘못 사용할 시에는 수십, 수백 배의 성능 저하를 가져올 수 있다. 여기서 최적의 경로란 일의 양을 적게하는 방법을 말한다.[1] 본 논문에서는 조인연산시 연결 컬럼(column)의 인덱스(index) 사용유무에 따라 처리 효율을 오라클 DBMS에서 제공하는 분석도구를 이용해서 여러 가지 질의 유형에 따라 효율성을 평가하고 성능개선에 유용한 자료로 활용하고자 한다.

2. SQL튜닝 방법

SQL튜닝이란 SQL문장에 대한 튜닝으로 데이타베이스를 효율적으로 사용하도록 SQL 문장을 수정하고 작성하는 것이다. 동일한 결과 값을 갖는 질의라도 SQL 문장의 형태에 따라 그 응답시간은 상이하게 다를 수 있다. 테이블에 인덱스가 적절하게 생성 되었어도 인덱스를 사용하지 못하는 SQL문장으로 질의한다면 데이타베이스 성능은 저하될 것이기 때문이다. 반면에 특정한 경우에는 인덱스를 사용하지 않는 것이 더 효율적이기 때문에 SQL튜닝을 통해 인덱스를 적절하게 선택하고 사용하는 것이 중요하다[2]. 특히 조인연산시 연결 컬럼의 인덱스 사용 유무에 따라 엄청난 수행속도 저하가 나타나므로 튜닝을 통해 조인되어야 할 각 집합의 처리범위와 가장 좁은 범위가 먼저 처리 할 수 있도록 SQL문장을 작성해야 한다[3]. 본 논문에서는 SQL의 실행 계획에 영향을 미치는 테이터베이스에서 수립된 통계정보를 이용하는비용기반 옵티마이저(COST-BASED OPTIMIZER)는 사용하지 않고, 데이터베이스에서 정한 조건의 우선순위에 의거하여 실행 계획을 수립하는 규칙기반 옵티마이져(RULE-BASED OPTIMIZER)를 이용하여, 조인연산시 연결고리 상태에 따라 나타나는 처리효율을 평가하고, SQL문 작성시 인덱스 생성 가이드와 적절한 힌트를 사용하여 수행속도를 높일 수 있는 방안을 제시한다.

3. 시험스키마와 질의

질의 유형에 따른 조인 연산에 대한 성능 평가하기 위해서 시험 데이터와 질의 유형은 다음과 같다.

3.1시험스키마

TAB1 테이블의 A_COL1 컬럼과 TAB2 테이블의 B_COL1 컬럼은 유일한 값의 분포를 갖고 있고, TAB1 테이블의 A_COL2컬럼과 TAB2 테이블의 B_COL2 컬럼은 10%의 유일한 값 분포를 갖으며, 각각 비클러스터 인덱스를 갖는다. 그리고 각 테이블에는 100,000건의 튜플을 갖는다.

[표1] 시험 스키마

TAB1
ID_CODE char(10)
A_COL1 char(10)
A_COL2 char(10)
:

TAB2
ID_CODE char(10)
B_COL1 char(10)
B_COL2 char(10)
:

3.2 시험질의

질의는 조인연산시 연결 컬럼에 따른 인덱스 사용유무와 질의 조건의 자료 선택률이 조건컬럼의 분포도 범위 10%와 50%에 따른 질의 유형으로 분류한다.

TAB1 테이블의 A_COL1 컬럼과 TAB2 테이블의 B_COL1컬럼을 조인하고, 이의 연결컬럼에 인덱스가 모두 존재할 때, 한쪽에만 인덱스가 존재할 때, 양쪽에 인덱스가 존재하지 않을때를 나누어서 질의한다.

[표2] 조인의 연결컬럼에 인덱스가 모두 존재할때

질의	조건
질의 1-1	선행, 후행테이블 각각 10%의 자료 선택률을 갖고 TAB1가 선행테이블일때
질의 1-2	선행, 후행테이블 각각 10%의 자료 선택률을 갖고 TAB2가 선행테이블일때
질의 1-3	선행테이블이 10%의 자료 선택률을 갖고, 후행 테이블은 50%의 자료 선택률을 갖을때.
질의 1-4	선행테이블이 50%의 자료 선택률을 갖고, 후행 테이블은 10%의 자료 선택률을 갖을때.

조인의 연결컬럼에 정상적으로 인덱스가 사용되었을 때는 선행테이블의 처리범위에 따라 수행속도를 떨어뜨릴 수 있다. [표2]은 검색되는 자료의 질의 선택률에 따라 수행 속도를 파악하며, 4개의 질의가 제안되었다.

[표3] 조인의 연결 컬럼에 한쪽만 인덱스가 있음

질의	조건
질의2-1	선행, 후행테이블이 각각 10%의 자료 선택률을 갖고 선행테이블에 인덱스가 없을때
질의2-2	선행, 후행테이블이 각각 10%의 자료 선택률을 갖고 후행테이블에 인덱스가 없을때
질의2-3	선행 테이블이 10%의 자료선택률, 후행테이블 50%선택률을 갖으며 선행테이블에 인덱스가 없을 때
질의2-4	선행 테이블이 10%의 자료선택률, 후행테이블 50%선택률을 갖으며 후행테이블에 인덱스가 없을 때
질의2-5	선행 테이블이 50%의 자료선택률, 후행테이블 10%선택률을 갖으며 선행테이블에 인덱스가 없을 때
질의2-6	선행 테이블이 50%의 자료선택률, 후행테이블10%선택률을 갖으며 후행테이블에 인덱스가 없을 때

조인의 연결 컬럼에 비정상적으로 인덱스가 사용되었을 때, 인덱스 사용유무에 따라 수행 속도를 떨어뜨릴 수 있다. [표3]는 검색되는 자료 선택률과 연결컬럼의 인덱스 사용유무에 따라 수행 속도를 파악하며, 6개의 질의가 제안 되었다.

[표4]조인의 연결 컬럼이 모두 인덱스가 없을때

질의	조건
질의3-1	선행, 후행테이블이 각각 10%의 자료 선택률을 갖을때
질의3-2	선행, 후행테이블이 각각 50%의 자료 선택률을 갖을때

조인의 연결 컬럼에 인덱스를 사용되지 않았을때는 FULL SCAN하여 수행 속도를 떨어 뜨릴 수 있다 . [표4]은 검색되는 자료 선택률에 따라 수행 속도를 파악하며 2개의 질의가 제안 되었다.

4. 시험결과 및 평가

실행환경은 4GB의 주 메모리를 가진 HP9000-V2500 시스템을 사용하였으며, 운영체제는 HP-UX 11.0을 사용하고, 데이터베이스는 ORACLE 8.0.5버전을 사용하였다. 분석도구는 오라클에서 제공하는 SQL_TRACE 와 EXPLAIN PLAN 유틸리티를 사용하였다. SQL_TRACE는 수행된 각종 SQL 작업에 대한 실행결과를 트레이스 파일로 생성시켜 주는 유틸리티이며, EXPLAIN PLAN는 사용자들이 SQL문의 액세스 경로를 확인하고 튜닝

할 수 있도록 SQL문을 분석하고 해석하여 실행계획을 수립한 후 실행계획을 PLAN_TABLE에 저장하도록 해주는 명령이다[4].

4.1 조인의 연결 컬럼에 인덱스가 모두 존재할 때 결과

[표5]연결 컬럼에 인덱스가 모두 존재 (단위:초)

질의	CPU	수행시간
질의 1-1	0.51	0.54
질의 1-2	0.51	0.45
질의 1-3	0.61	0.63
질의 1-4	2.24	2.60

조인연산시 양쪽 모두에 인덱스가 존재하는 경우 선행테이블의 자료처리범위에 따른 수행속도를 평가한다. 자료 처리범위가 동일할 때는 어느 방향으로 연결해도 수행속도는 별 차이 없고, 동일하지 않을 때에는 먼저 처리범위를 줄여 주는 테이블이 먼저 수행되면 수행속도는 효율적이다.

4.2 조인의 연결 컬럼에 인덱스가 한쪽만 존재할 때 결과

[표6]연결 컬럼에 인덱스가 한쪽만 존재 (단위:초)

질의	CPU	수행시간
질의 2-1	0.50	0.46
질의 2-2	938.34	938.76
질의 2-3	0.66	2.08
질의 2-4	5050.21	5053.32
질의 2-5	2.23	6.14
질의 2-6	4863.93	4864.75

조인시 연결 컬럼에 한쪽만 인덱스가 있을때 자료 처리 범위가 동일할 경우에는 인덱스가 없는 테이블이 먼저 수행 되었을때가 인덱스가 있는 테이블이 먼저 수행한 것 보다 훨씬 수행속도가 효율적이며, 자료 선택률이 많고 인덱스가 없는 테이블이 나중에 처리 될 수록 엄청난 수행속도를 떨어 뜨렸다.

4.3 조인의 연결 컬럼에 모두 인덱스가 존재하지 않을때 결과

[표7]연결컬럼에 모두 인덱스가 존재하지 않음(단위:초)

질의	CPU	수행시간
질의 3-1	894.72	896.66
질의 3-2	23819.12	23851.58

연결 컬럼 모두에 인덱스가 없을 때는 엄청난 수행시간이 걸렸으며, 자료처리범위가 많을수록 수행속도는 훨씬 많이 떨어뜨렸다.

세가지 질의 유형에 따른 결과에 의하면 조인의 연결 컬럼에 인덱스가 존재하고, 먼저 처리되는 테이블의 자료 선택률이 적을 때 일수록 수행속도는 효율적이며, 인덱스가 존재하지 않을때는 인덱스가 존재하지 않는 테이블을 먼저 처리하면 수행속도를 높일 수 있다는 것을 알 수 있다.

5. 결론 및 향후 연구과제

데이터베이스 시스템에서 프로세스가 과도하게 CPU를 점유하는 경우는 주로 비효율적인 SQL이 원인인 경우가 많다. 응용프로그램 작성시 소량의 데이터를 처리했을 때는 그 처리 방법이 아무리 비효율적이라 하더라도 바로 원하는 결과를 얻을 수 있어 문제의 심각성을 못 느끼지만, 대량의 데이터에서는 성능상의 한계를 느끼게 된다. 데이터베이스에 내장된 옵티마이저(Optimizer,최적기)가 수립된 통계정보를 이용하여, 최적의 길을 알아서 처리 해 주지만, 아무리 잘 만들어진 옵티마이저라 할지라도 종합적인 판단을 알고리즘으로 해결할 수 는 없으며, 사람의 판단을 필요로 한다[5]. 따라서 본 논문에서는 옵티마이저가 실행계획을 잘못 세우거나 더 좋은 경로를 알고 있을 때 사용자가 조인 SQL문을 작성시 적절한 인덱스 사용과 자료처리 범위를 줄일 수 있도록 선행테이블을 먼저 수행하도록 힌트를 사용하는 가이드로 제공한다. 향후 과제로서는 데이터 연결시 다양한 방법으로 조인의 횟수를 줄일 수 있는 방법이 요구된다.

참고문헌

[1]박태욱, SQL 튜닝의 몇가지 비결, 웨어벨리 http://www.warevalley.com/forum/forum_consul_01.asp
[2]안기덕,오정석,이상호, 데이타베이스 튜닝도구 개발, 한국정보처리학회 논문지 제7권 제11호, p3312, 2000.11
[3]이화식, 대용량 데이타베이스 솔루션 I, 대청, p91, 1996
[4]ORACLE Inc. Oracle8 Tuning Guide, p4-6, 2002
[5]이화식, 대용량 데이타베이스 솔루션 II, 대청, p1-2, 1998