

# Oracle DBMS 질의 최적화기 한계 극복을 위한 진단 도구 구현

서민선, 황윤재, 이상원  
성균관대학교 정보 통신공학과  
e-mail:wonlee@ece.skku.ac.kr  
{kingsms, somy1545}@skku.edu

## The Implementation of Tool for overcoming the limit of Oracle optimizer

Sang-Won lee      Min-Sun Seo      Yoon-Jae Hwang  
Dept. of Information and communication Engineering,  
Sungkyunkwan University

### 요 약

관계형 DBMS 질의 최적화기는 기본적으로 통계정보(statistics), 선택도(selectivity), 카디널리티(cardinality), 비용(cost) 순으로 특정 실행 계획(Execution Plan)의 수행비용을 예측한다. 질의 최적화기가 최적의 실행 계획을 선택하는데 있어서 정확한 통계정보 유지와 올바른 선택도 그리고 카디널리티 예측이 무엇보다 중요하다. 이 논문에서는 관계형 DBMS의 질의 최적화기가 비용을 계산하는 과정에서 발생할 수 있는 오류의 종류와 원인을 밝히고 각각의 오류에 대한 해결 방안을 진단 도구를 통하여 제시하도록 하겠다. 질의 최적화기의 오류로 인해 발생하는 잘못된 선택도와 카디널리티 예측을 인지하고 사용자에게 적절한 해결책을 제시한 후 실행 계획이 어떻게 바뀌었는지, 성능이 얼마나 향상되었는지를 확인하기 위해 진단 도구를 개발하고 그 결과를 분석하였다. 실험을 통해 본 논문에서 제시하는 질의 최적화기의 오류로 인한 문제를 가시적으로 확인할 수 있었으며, 부정확한 통계정보 유지와 잘못된 선택도 예측으로 인해 발생하는 문제가 어느 정도 해결된 것을 확인할 수 있었다.

### 1. 서론

정보 기술의 발달과 함께 모든 정보는 DBMS에 저장되고 있으며 저장되는 데이터의 크기 또한 기하급수적으로 늘어나고 있다. 또한 DBMS를 이용하는 사용자는 빠르고 정확하게 데이터의 검색 결과를 요구하고 있다. 이러한 환경에서 관계형 DBMS(Database Management Systems)의 두뇌에 해당하며 데이터의 검색 시간을 결정하는 질의 최적화기(Query Optimizer)의 역할은 무엇보다 중요하다. 질의 최적화기는 가장 짧은 시간에 질의 결과를 도출하기 위한 질의 실행 계획을 생성하고 DBMS는 이를 기반으로 데이터를 처리하는데 대부분의 DBMS 질의 최적화기는 항상 최선의 실행 계획을 수립하지 못한다.

본 논문에서는 대표적인 DBMS중의 하나인 오라클(Oracle) 질의 최적화기가 최선의 실행 계획을 수립하지 못하는 경우를 파악하고, 이에 대한 오류의 원인과 해결 방안을 모색하여 진단도구를 통한 자동적인 오류 해결 방안을 제시하고 있다.

질의 최적화기의 동작 방식은 크게 RBO(Rule-Based Optimization)와 CBO(Cost-Based Optimization)모드 두 가지로 볼 수 있는데 본 논문에서는 CBO 모드의 질의 최적화기에 대해서만 논의하겠다.

본 논문의 2장에서는 비용 기반 질의 최적화기의 오류와 해결방안에 대해 분석하고 3장에서는 진단 도구를 통한 한계 극복 방안의 기본 원리에 대해서 소개한다. 4장에서는 제안하는 방법에서 미해결된 사항들을 소개하고 향후 연구 방향을 제시하고자 한다.

\* 본 연구는 한국과학재단 목적기초연구(R05-2003-000-11943-0) 지원으로 수행되었음.

### 2. 비용 기반 질의 최적화기의 오류

비용 기반 질의 최적화기는 통계정보를 기반으로 선택도와 카디널리티를 계산하는 내부 공식을 가지

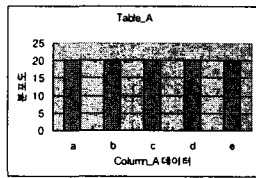
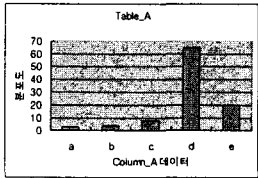
고 있다. 이 내부 공식들은 다음 세 가지 가정에 기반하고 있다.[2]

- 균등분포(Uniform distribution) 가정
- 조건 식 상호 독립(Predicate Independence) 가정
- 조인 독립(Join Independence)

이러한 가정은 대부분의 경우에는 성립을 하지만 모든 경우에 성립하지 않는 불완전한 가정이다.

### 2.1 데이터 균등 분포 가정 및 해결 방안

질의 최적화기는 데이터 디튜너리의 통계 정보를 기반으로 선택도(Selectivity)와 카디널리티(Cardinality)를 산정 할 때 첫 번째, 균등 분포가정을 통해 실행 계획을 수립한다. 균등 분포 가정이란 DBMS가 히스토그램 단계가 아닌 기본 단계의 통계 정보를 유지하고 있고 테이블의 데이터가 (그림 1)과 같이 분포되어 있다면 DBMS의 질의 최적화기는 데이터의 분포도를 "전체 튜플 수 \* 1/컬럼의 서로 다른 값의 개수"로 가정한다. 따라서 (그림 2)와 같이 데이터가 균등하게 분포되어 있다고 가정하는 것이다.



(그림 1) 데이터가 저장된 실제 분포도 (그림 2) 질의 최적화기가 가정한 데이터의 분포도

*select \* from Table\_A where Column\_A = 'a'* 질의 경우에 질의 최적화기는 이 가정에 따라 전체 테이블 검색(Full table scan)을 선택할 확률이 아주 높다. 그러나 이 경우에 인덱스 검색(Index scan)을 이용하면 테이블 검색 속도는 향상 될 것이다. 이 오류에 대한 해결 방안은 통계정보의 수준을 히스토그램 단계로 만들어 주고 주기적으로 테이블을 분석하여 통계 정보 수준을 최적의 상태로 유지하면 해결된다. (그림 3)은 이 문제를 해결하기 위해 진단 도구에서 구현한 알고리즘이다.

```

알고리즘 1.
//오차 = abs (cardinality-last_output_rows) / last_output_rows
if(조건식이 한 개이고 오차가 0.05보다 크다)
{
    if(통계치가 오래 되거나 유효하지 않다.)
        테이블에 대해 ANALYZE 수행 //last_analyzed이용
    else if(데이터가 쏠려(skewed) 있다)
        해당 컬럼에 대해 히스토그램 생성
}
    
```

### 2.2 조건식 상호 독립(Predicate Independence) 가정 및 해결 방안

조건 식 상호 독립 가정은 하나의 테이블에 대해 주어진 조건 절의 각 조건 식이 서로 독립적이라는 것이다. 질의 *select \* from emp where job\_title='부장' and salary<4000*의 경우, *job\_title='부장'*과 *salary<4000*의 정확한 선택도가 각각 0.1, 0.4라고 했을 때, 질의 최적화기는 두 개의 조건식이 서로 독립적이라는 가정 아래 결합 선택도를 0.1 \* 0.4 즉, 0.04 로 계산을 하게 된다 [3]또는 [4]. 하지만 부장이면서 연봉이 4000이하인 경우는 거의 없기 때문에 실제 선택도는 0에 가까울 것이다. 살펴본 바와 같이, 이 가정에 위배될 경우, 즉 실제 데이터 분포 상 서로 밀접한 상관관계가 있는 두 칼럼에 대한 선택도를 구할 때, 질의 최적화기는 잘못된 선택도를 산정 하게 되고 따라서 잘못된 카디널리티와 예상비용을 산정 하게 된다. 이어서 이에 대한 해결책을 제시 하겠다.

(그림 3)과 같은 질의에서, 조건 식 P1(*prod\_subcategory = 'Shoes - Women'*)과 P2(*prod\_pack\_size = 'wooden case'*)에 대해 각각 정확한 통계정보를 유지시킨 다음에(i.e. 각각의 칼럼에 대해 히스토그램을 만들어 준다.) 질의 최적화기가 예측한 카디널리티와 실제 카디널리티를 비교한 결과를 (그림 4)에서 나타내었다.

```

SELECT count(*)
FROM products
WHERE prod_subcategory='Shoes - Women' and
      prod_pack_size='wooden case';
    
```

(그림 3)

Dynamic Performance View		Query Result	Explain Plan
Operation	COST	CARDINALITY	LAST_OUTPUT_ROWS
1   0 SORT AGGREGATE		1	1
2   P2 TABLE ACCESS FULL PRODUCTS	37	57	207

(그림 4) 진단 도구 분석 결과

각각의 조건식에 해당하는 칼럼에 대해 정확한 통계 정보를 유지 시켰음에도 불구하고 (그림 4)에서의 결과처럼 질의 최적화기가 예측하는 카디널리티와 실제 카디널리티가 차이가 많이 나는 경우, 조건 식 상호 독립의 가정에 위배되었다고 판단할 수 있다. 이 경우 사용자가 데이터를 직접 분석하여 질의 최적화기에게 더 좋은 실행 계획을 선택하도록 힌트(hint)기능을 이용하거나 stored outline을 이용하는 방법이 있을 수 있다. 또한 Oracle9iR2 Database 같은 경우 동적 샘플링(Dynamic Sampling)이라는

기능을 제공해서 좀 더 정확한 선택도 계산을 가능하게 하고 있다 [7]. (그림 5)은 Oracle9iR2 Database에서 제공하는 동적 샘플링을 이용하여 오차를 수정한 예를 보여주고 있다.

Operation	COST	CARDINALITY	LAST_OUTPUT_ROWS
1 53 0 SORT AGGREGATE		1	1
2 50 0 SORT AGGREGATE		1	1
3 50 7 TABLE ACCESS SAMPLE PRODUCTS	35	1,707	1,844
4 50 6 TABLE ACCESS FULL PRODUCTS	37	107	207

(그림 5) 진단 도구 분석 결과

(그림 4)에서 질의 최적화기가 실제 카디널리티에 비해 4배나 적은 값을 예측을 했으나, (그림 5)에서 처럼 동적 샘플링을 통해 질의 최적화기가 187로 실제 값(207)과 거의 비슷한 카디널리티를 예측한 것을 확인할 수 있다. (여기서는 샘플링 값을 2로 주었을 때 결과이며, 값을 증가시키면 정확도가 높아질 가능성도 커진다). 이 문제를 해결하기 위해 진단 도구에서 구현한 알고리즘은 (그림 6)와 같다.

```

알고리즘 2.
if (한 테이블에 조건식이 2개 이상이고 오차가 0.05보다 크다)
{
    if (통계치가 오래 되었다)
        테이블에 분석(Analyze) 수행
    do {
        if(조건식이 걸려 있는 컬럼이 skewed되었다)
            해당 컬럼에 히스토그램 생성
        until (조건식에 해당하는 모든 컬럼이 skewed 여부를 검사 받을 때까지)
        if(재실행 후 여전히 오차가 5%이상이면)
        {
            while(오차가 5%이상)
            {
                sample_size=1;
                동적 샘플링을 한다. //힌트(hint)사용
                sample_size ++;
            }
        }
    }
}
    
```

(그림 6) 조건 식 상호 독립 가정의 오류 극복 방안

### 2.3 조인 독립(Join Independence) 가정 및 해결 방안

조인 독립은 조인되는 두 테이블 T1과 T2에 대해, T1의 한 튜플이 T2의 모든 튜플과 똑같은 확률로 조인된다고 가정하는 것이다. 예를 들어, 질의 select \* from emp, dept where emp.deptno = dept.deptno의 선택도는 1/MAX(emp.dept의 NDV, dept.deptno의 NDV)로 계산이 된다(null 값이 없을 때) [5]. 극단적인 경우는, 아래 <표2>에서처럼 양쪽 컬럼 모두 특정 값이 많이 나타나는 경우로, 이들이 조인될 때는 조인 조건의 선택도는 1에 가까울 것이다. 즉, 질의 최적화기는 조인 독립 가정이 지켜진다는 가정 하에 선택도를 1/3으로 예측을 하게 되지만, 실제로 18/36, 즉 1/2이 된다.

<표 2>

...	deptno	deptno	...
	10	10	
	10	10	
	10	10	
	10	10	
	20	20	
	30	30	

다음은 위의 조인 독립 가정에 대한 해결책을 제시한다. (그림 7)의 질의에서 조인이 일어나는 칼럼들(T1.a, T2.a)에 대해 각각 정확한 통계정보를 유지시켰다고 가정할 때, (그림 8)과 같은 결과가 나왔다고 가정하자.

```

SELECT count(*)
FROM join_not_uniform T1, join_not_uniform T2
WHERE T1.a = T2.a ;
    
```

(그림 7)

Operation	CARDINALITY	LAST_OUTPUT_ROWS
1 95 0 SORT AGGREGATE	1	1
2 95 83 HASH JOIN	5	909
3 95 0 TABLE ACCESS FULL JOIN_NOT_UNIFORM	2	100
4 95 0 TABLE ACCESS FULL JOIN_NOT_UNIFORM	2	100

(그림 8)

해시(Hash) 조인이 일어 날 때, 질의 최적화기가 예측한 카디널리티와 실제 카디널리티가 9배 정도 차이가 난다. 이 경우 조인 독립 가정을 위배했다고 판단할 수 있다. 원래 조인 독립 가정에 위배되어 발생하는 오차는 실제 데이터 분포를 확인한 후 수동으로 찾아야 하고, 이를 바탕으로 사용자가 힌트(Hint) 기능을 이용하여 튜닝(Tuning)을 해 주어야 한다. 하지만, 여기서 제시하는 진단 도구에서는 비용 산정 모듈을 이식하여 이를 어느 정도 자동화하였다. 기존의 질의 최적화기가 가지고 있는 비용 산정 모듈과 다른 점은 기존의 질의 최적화기의 비용 산정은 예상 카디널리티를 이용하여 비용을 계산하는 반면에 이 응용프로그램에 이식된 비용 산정 모듈은 이미 한 번 수행하여 얻은 정확한 실제 카디널리티를 이용하여 비용을 산정 하는 것이다. 그래서 최적의 조인 방법을 찾아내서 힌트(hint)기능으로 계획을 바꿀 수 있도록 한다. 이와 관련된 알고리즘은 (그림 9)와 같다.

**알고리즘 3.**

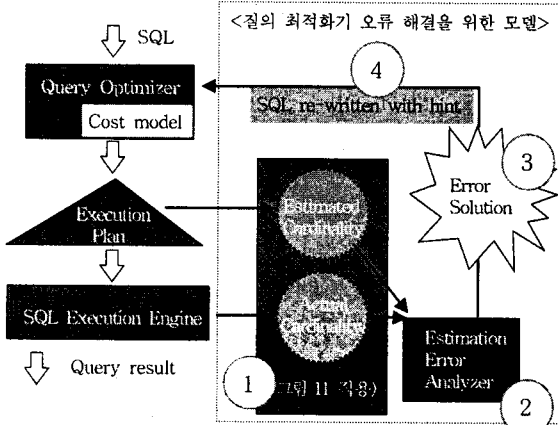
```

if (오차가 5%이상이며 실행 계획의 연산(operation)이 Join일 때)
{
    if (통계치가 오래 되거나 유효하지 않다.)
        테이블에 대해 ANALYZE 수행
    if(재실행 후 여전히 오차가 5% 이상이면)
    {
        새로운 비용 산정 모듈을 호출하여 최적의
        조인 방법을 찾는다.
    }
}
    
```

(그림 9)

**3. 진단 도구를 통한 한계 극복 방안의 기본 원리**

질의 최적화기의 한계의 근본적인 원인은 선택도 (Selectivity)와 카디널리티(Cardinality)산정에 있다. 본 논문에서 제시하는 진단도구의 기본 원리는 (그림 10)의 “질의 최적화기 오류 해결을 위한 모델”과 같이 먼저 Oracle에서 제공하는 동적 뷰<표 3>를 이용하여 선택도와 카디널리티의 오차를 발견(그림 5)하고 오차 수정에 따라 좀더 나은 테이블 접근 방법(access method), 조인 순서(join ordering), 조인 방법(join method)등의 실행 계획을 도출(그림 5의 2와 3)했을 때 이 실행 계획을 수립할 수 있도록 실행 계획 변경 방법을 제공 하는데 목적이 있다.



(그림 10)

<표 3> 선택도와 카디널리티의 오차 검색에 유용한 동적 성능 뷰

뷰	유용한 컬럼
V\$SQL	SQL_TEXT, HASH_VALUE, ADDRESS
V\$SQL_PLAN_STATISTICS_ALL	CARDINALITY, LAST_OUTPUT_ROWS, COST, LAST_CR_BUFFER_GETS

```

SELECT
lpad(' ', 4 * (level - 1)) || vs.operation
|| ' ' || vs.options
|| ' ' || vs.object_name AS operation
, vs.cardinality , vs.last_output_rows
, vs.filter_predicates, vs.access_predicates, id
, vs.parent_id
FROM v$sql v, v$sql_plan_statistics_all vs
WHERE v.sql_text = '&search_sql'
and vs.address = v.address
and vs.hash_value = v.hash_value
start with vs.id=1
connect by prior vs.id = vs.parent_id
and prior vs.address = vs.address
order by id;
    
```

(그림 11) 선택도와 카디널리티 오차 검색 질의

**4. 결론 및 향후 연구**

본 논문에서 이야기한 진단 도구는 질의 최적화기의 균등분포 가정과 조건 식 상호 독립 가정에 대해서만 오류의 해결책을 제시하였다. 그러나 이것은 병산의 일각일 뿐이며 현재 질의 최적화기가 내포하고 있는 다른 한계성 즉, 조인 독립 가정과 실행 환경에 대한 가정(e. g. 중첩 루프 조인 시 해당 데이터 및 인덱스 블록은 디스크로부터 읽어야 한다는 가정과 정렬 합병 또는 해시 조인에 필요한 메모리 크기에 대한 가정)등의 한계성은 아직 본 논문에서 미 해결책 사항으로 남아있으며 향후 이에 대한 뚜렷한 해결책을 제시하고 이를 진단 도구를 통해 구현하는 것이 연구과제로 남아 있다.

**참고문헌**

1. Banchong Harangsri, John Shepherd, Anne H. H. Ngu: *Query Size Estimation Using Systematic Sampling* proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications, 1996. Kyoto, Japan.
2. Wolfgang Breitling, Centrex Consulting Corporation : *Fallacies of the cost based optimizer*
3. Wolfgang Breitling: *A Look under the Hood of Cbo: The 10053 Event*. www.centrexcc.com
4. 35934.1: *Cost Based Optimizer - Common Misconceptions and Issues*. www.metalink.oracle.com
5. 68992.1: *Predicate Selectivity*. metalink.oracle.com
6. Wolfgang Breitling, Centrex Consulting Corporation: *What is new in the CBO - And the 10053 event trace - in ORACLE 9i*
7. Oracle 9i Database Performance Tuning Guide and Reference. 2002: Oracle Corporation.