바다-II에서 조기 프로젝션의 설계와 구현

김정아[O], 박영철
경북대학교 컴퓨터과학과
aya07@hanmail.net[O], ycpark@knu.ac.kr

Design and implementation of the early projection in BADA-II

Jeong A Kim[0], Young Chul Park
Dept. of Computer Science, Kyungpook National University

요 약

기존의 바다-II는 늦은 프로젝션(late projection) 기법을 사용하기 때문에 중간 임시 결과 파일의 디스크 쓰기와 디스크 읽기에 많은 디스크 자원, 메모리 자원, 그리고 CPU 시간이 소모되었다. 본 논문은 바다-II에 조기 프로젝션(early projection) 기법의 설계와 구현 그리고 그에 따른 성능 향상을 보인다.

1. 서 론

조인과 같이 질의 수행 과정에 중간 결과가 생성되는 경우, 그의 처리는 구체화(materialization) 또는 파이프라이닝(pipelining) 기법으로 수행될 수 있다. 구체화 기법은 다음 연산의 수행을 위하여 중간 결과를 임시 파일에 저장하며, 파이프라이닝 기법은 중간 결과의 저장 없이 바로 다음 연산에 결과를 넘겨준다. 늦은 프로젝션은 모든 조인 연산 후에 프로젝션 연산을 수행하므로 최종 결과에는 필요하지 않은 칼럼들을 중간 결과 파일에 저장하느라 많은 비용이 든다. 칼럼의 개수가 많을수록, 결과 투플의 개수가 많을수록 비용은 더 커진다. 바다-II의 기존 버전은 조인의 수행 시 늦은 프로젝션과 구체화 기법을 사용하였다. 본 논문은 바다-II에 조기 프로젝션 기법을 어떻게 구현하였는지 그리고 어느 정도의 성능 향상을 이루었는지를 보인다. 본 논문의 나머지 부분의 구성은 다음과 같다. 제 2장에서 조기 프로젝션 수행을 위한 질의 실행 계획을 제시하고, 제 3장에서 조기 프로젝션의 성능 평가 결과를 보인다. 제 4장에서 본 논문의 결론을 맺으며 향후 연구 과제를 기술한다.

2. 조기 프로젝션 수행을 위한 질의 실행 계획

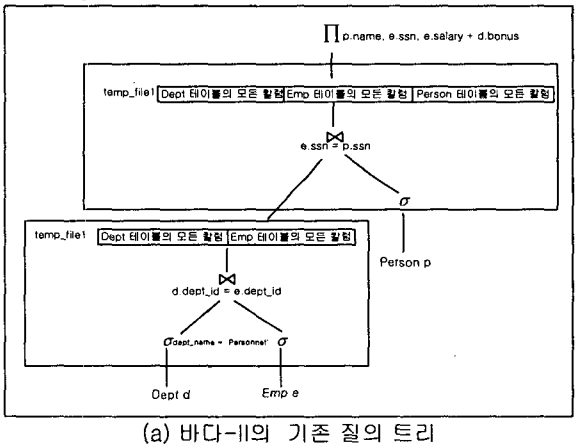
클라이언트가 서버에 요구한 질의문은 파싱, 정당성 검사를 거쳐 최적화 과정에서 설정한 질의 실행 계획에 따라 수행된다. 본장은 조기 프로젝션의 수행을 위하여 질의 최적화 과정에서 설정할 질의 실행 계획에 대하여 바다-II를 기준으로 설명한다.

2.1 질의 트리

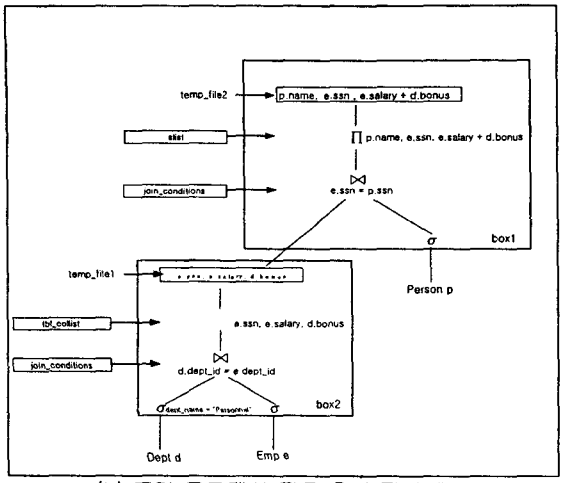
질의 트리는 관계 대수식에 대응되는 트리자료 구조이다. 아래의 예를 통하여 바다-II 질의 트리를 설명한다.

[예제 1] SELECT p.name, e.ssn, d.salary + d.bonus
FROM Person p, Emp e, Dept d
WHERE p.ssn = e.ssn AND
e.dept_id = d.dept_id AND
d.dept_name = ' Personnel' ;

[예제 1]에 대한 질의 트리는 그림 1과 같다.



(a) 바다-II의 기존 질의 트리



(b) 조기 프로젝션 적용 후의 질의 트리

그림 1. 예제 1에 대한 질의 트리

그림 1-a는 바다-II의 기존 질의 트리로서 조인 연산의 수행 후에 프로젝션 연산을 수행한다. 따라서 최종 결과를 도출해내는데 필요하지 않은 칼럼들을 저장하고, 읽는데 오버헤드가 따른다. 그림 1-b는 바다-II에 조기 프로젝션을 도입한 후의 질의 트리로서 bottom up 방식으로 조인을 수행하면서 다음 연산에 필요한 칼럼들과 최종 project 시에 필요한 칼럼들만을 project 하여 중간 임시 결과 파일에 저장한다. 조인 수행 중에 최종 결과를 도출해내는데 필요하지 않은 칼럼들을 버림으로써 바다-II의 기존의 늦은 프로젝션에 드는 오버헤드를 줄인다.

2.2 질의 실행 계획

최적화 과정에서 질의 최적화기는 질의를 효율적으로 수행하기위한 방법을 찾아내어 질의 실행 계획을 만든다. 바다-II의 질의 실행 계획은 left skewed 트리 형태이다. 그 형태를 그대로 유지하면서 조기 프로젝션에 필요한 자료를 추가하여 조기 프로젝션을 수행할 수 있도록 설정한다. 조기 프로젝션의 수행을 위하여 TABLE_PLAN 구조체에 추가된 정보는 다음과 같다.

- slist : select 리스트이다. 질의 실행 계획의 root 노드의 경우에만 NOT NULL이다.
- with_aggregation : root 노드의 slist가 aggregation을 가질 경우에만 그 node의 with_aggregation이 TRUE로 설정된다.
- join_conditions : 조인 정보를 가진다.
- join_result_dtd->tbl_collist : 중간 결과 칼럼 정보를 가진다.

최적화기가 조인 수행 방법으로 중첩 루프 조인 알고리즘을 선택할 경우, 그림 1-b의 box1에 필요한 정보는 <오른쪽 테이블에 대한 selection 연산 정보, 조인 조건, select 리스트, tbl_collist>이다. 이는 그림 2의 질의실행 계획의 root 노드가 가지는 정보와 동일하다. 그림 1-b의 box2에 필요한 정보는 <오른쪽 테이블에 대한 selection 연산 정보, 조인 조건 , tbl_collist> 이다. 이는 그림 2의 질의 실행 계획의 내부 노드가 가지는 정보와 동일하다.

그림 1-b의 질의 트리를 질의 실행 계획으로 변환하면 그림 2와 같다.

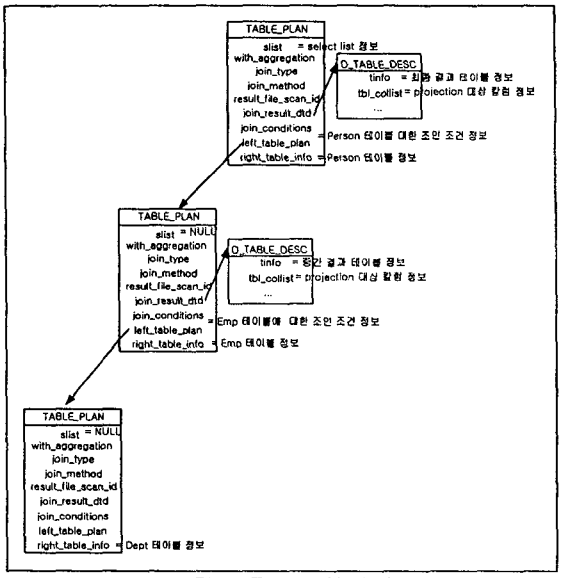


그림 2. 질의 실행 계획

2.4 조기 프로젝션 수행을 위한 칼럼 정보 설정 알고리즘

left_table_plan이 NULL이 아닐 경우에 다음 정보를 설정한다.

- Root 노드 : slist에 select 리스트를 연결한다.
- Second 노드 : root 노드의 slist, 조인 조건, 필터 조건에 속하는 각 칼럼들 중 root 노드의 오른쪽 테이블에 속하지 않는 칼럼들의 리스트를 join_result_dtd->tbl_collist에 연결한다. (root 노드의 조인연산, 필터 연산, 최종결과 칼럼 생성시에 오른쪽 테이블에 속하지 않는 칼럼값은 중간 결과 레코드에서 읽어와야 한다. 그러기 위해서는 전 단계 프로젝션 연산시 위 세 조건에 속하는 칼럼들을 모두 프로젝션 하여야한다. 그 프로젝션 칼럼들의 정보를 tbl_collist가 가지는 것이다.
- Third 노드 이상 : 부모 노드의 tbl_collist, 조인 조건, 필터 조건에 속하는 각 칼럼에서 부모 노드의 오른쪽 테이블에 속하지 않는 칼럼들의 리스트를 join_result_dtd->tbl_collist에 연결한다. (그 이유는 second 노드와 동일하다.)

2.5 input, output 칼럼 정보 설정 알고리즘

조기 프로젝션을 수행할 경우, 칼럼값을 읽어오기 위해서는 중간 결과 레코드 또는 base 테이블 내에서의 해당 칼럼의 위치 정보를 알고 있어야 한다. 그 정보를 input 칼럼 정보라고 한다. 읽어온 칼럼을 통하여 새로운 결과 칼럼을 생성 시에 새로운 결과에서 칼럼의 위치 정보를 알고 있어야 한다. 이 정보가 output 칼럼 정보이다. Base 테이블에서 칼럼의 위치는 고정적이지만, 중간 결과 레코드 내에서의 칼럼 위치는 유동적이다. 따라서 중간 결과 레코드에서 읽어올 칼럼은 이전 프로젝션의 output 칼럼 정보를 보고 input 칼럼 정보를 설정하여야 한다.

Input, output 칼럼 정보는 두단계 설정 알고리즘으로 설정된다. (left_table_plan의 left_table_plan이 NULL이 아닌 경우에 다음을 수행한다.)

표 1. input, output 칼럼 정보의 두단계 설정 알고리즘

	slist	tplan->join_result_dtd->tbl_collist	tplan->join_conditions	tplan->right_table_info의 필터 조건
1st Pass	input_column_info (base table)	output_column_info, input_column_info(base table)	input_column_info (base table)	input_column_info (base table)
2nd Pass	input_column_info (중간 결과 table)	input_column_info (중간 결과 table)	input_column_info (중간 결과 table)	input_column_info (중간 결과 table)

- 1st Pass : 각 칼럼들의 input 칼럼 정보는 base 테이블 내에서의 칼럼 정보를 설정하고, output 칼럼 정보는 중간 결과 레코드에서 해당 칼럼이 저장될 위치 정보를 설정한다.
- 2nd Pass : 중간 결과 파일에서 읽어와야할 각 칼럼들의 input 정보를 왼쪽 테이블의 tbl_collist의 output 정보를 보고 설정한다.

2.6 질의 실행

질의 실행 계획은 bottom up 방식으로 수행된다. Base 테이블 또는 중간 결과 화일에 대하여 selection 연산을 수행하고, 조인 조건을 만족하면 join_result_dtd->tbl_collist 정보에 따라 중간 결과 투플을 생성하여 중간 결과 파일에 저장한다. Root 노드에서는 조인 조건을 만족하는 투플을 slist의 정보에 따라 최종 결과 투플을 생성하여 최종 결과 파일에 저장한다.

3 성능 평가

최적화에 의해 조인 수행 방법을 중첩 루프 조인 알고리즘으로 설정하였을 경우, 늦은 프로젝션과 조기 프로젝션의 성능을 평가하였다. 조인 수행 시 인덱스가 존재할 경우, 인덱스를 사용한다. 테이블 정보는 다음과 같다. 각 테이블의 투플 수는 10000 개이다.

- Person (ssn, name, age, sex, city),
  PK = ssn
- Emp (ssn, emp_id, dept_id, positions, salary)
  PK = ssn, unique = emp_id, FK = dept_id
- Dept(dept_id, dept_name, dept_chair, bonus)
  PK = ssn, unique = dept_name
- Family(ssn, emp_id, marital_status, family_num)
  PK = ssn, unique = emp_id
- Univ(ssn, emp_id, univ, major, degree, grades)
  PK = ssn, unique = emp_id

각 조인에 따른 예제는 다음과 같다.

[예제 2] 1 조인에 대한 예
```
SELECT e.ssn, p.name
FROM Emp e, Person p
WHERE e.ssn = p.ssn;
```

[예제 3] 2 조인에 대한 예
```
SELECT e.ssn, p.name, d.dept_name
FROM Emp e, Person p, Dept d
WHERE e.ssn = p.ssn AND
      e.dept_id = d.dept_id;
```

[예제 4] 3 조인에 대한 예
```
SELECT e.ssn, p.name, d.dept_name, f.marital_status
FROM Emp e, Person p, Dept d
WHERE e.ssn = p.ssn AND
      e.dept_id = d.dept_id;
```

[예제 5] 4 조인에 대한 예
```
SELECT e.ssn, p.name, d.dept_name,
      f.marrital_name, u.major
FROM Emp e, Person p, Dept d, Family f, Univ u
WHERE e.ssn = p.ssn AND
      e.dept_id = d.dept_id AND
      e.ssn = f.ssn AND
      e.ssn = u.ssn;
```

위의 각 예제 수행 시 소요되는 시간은 표 2와 같다.

표2. 늦은 프로젝션과 조기 프로젝션 시 소요되는 시간

	늦은 프로젝션	조기 프로젝션	비고	성능 향상
1조인	10.39 sec	9.85 sec	-0.46 sec	4.43%
2조인	20.67 sec	18.44 sec	-2.23 sec	10.79%
3 조인	31.79 sec	27.27 sec	-4.52 sec	14.22%
4 조인	41.23 sec	35.93 sec	-5.30 sec	12.85%

조기 프로젝션을 도입한 경우, 기존의 늦은 프로젝션보다 성능이 우수하였다. 조인의 수가 많을수록 조기 프로젝션과 늦은 프로젝션 시 소요되는 시간차가 커졌다. 이는 늦은 프로젝션 시 조인의 수가 늘어날수록 중간 결과 투플의 크기가 커지므로 중간 결과 파일에 저장할 데이터의 양도 그 만큼 많아지기 때문이다. 4개 조인의 경우, 조기 프로젝션 시 소요되는 시간이 3개 조인의 경우보다 많이 감소되었지만, 성능 향상 정도는 낮아졌다. 그 이유는 실제 조인에 소요되는 시간의 증가하는 시간에 비해 조기 프로젝션으로 감소된 시간이 적기 때문이다.

4 결론 및 향후연구

최적화 과정에서 중첩 루프 조인 알고리즘으로 조인을 수행했을 경우, 조인의 숫자에 따라 조기 프로젝션 적용 후 작게는 4.43%에서 크게는 14.22%까지 성능 향상을 보였다. 이는 임시 결과 파일에 중간 결과 투플을 저장하고 읽어들이는데 드는 비용이 적게 들었기 때문이다. 바다-II에 조기 프로젝션 기법을 도입한 이유는 최종적으로 파이프라이닝을 설계하고 구현하기 위함이다. 본 논문을 바탕으로 향후 바다-II에서 파이프라이닝을 설계하고 구현하고자 한다.

5. 참고 문헌

[1] Ramez Elmasri and Shankant B, Navathe, " Fundamentals of Database Systems," 2000, Addison Wesley, Inc.

[2] Abraham Silberschatz, Henry F. Korth, S.Sudarshan, " DATABASE SYSTEM CONCEPTSs," 1997, McGraw-Hill Companies, Inc.