

Access Paths for Joins

Tom Kelliher, CS 318

Apr. 24, 2002

1 Administrivia

Announcements

Assignment

Read 15.1.

Group assignment due beginning of class 5/1: demonstrate that each of your project's tables are in BCNF or 3NF. For any tables that are not in these normal forms, provide a decomposition that is in dependency-preserving BCNF (preferably) or 3NF. Post your demonstration to your group web site.

From Last Time

Access paths: sorting, selection.

Outline

1. Finish up select processing from last time.
2. Access paths for joins.

Coming Up

Transactions — Isolation.

2 Access Paths for Joins

Three of many paths:

1. Nested loops join.
2. Block-nested loops join.
3. Hash join.

Preliminaries:

1. Two tables:
 - (a) Table 1 has $P_1 = 1,000$ pages and $R_1 = 10,000$ rows.
 - (b) Table 2 has $P_2 = 100$ pages and $R_2 = 1,000$ rows.
2. We are computing $T_1 \bowtie_{A=B} T_2$ where A is in Table 1 and B is in Table 2.
3. When counting page accesses, we don't count writes.

2.1 Nested Loops Join

The simplest method:

```
for each tuple t1 in Table 1
  for each tuple t2 in Table 2
    if t1.A = t2.B
      output <t1, t2>
```

How many pages accesses?

Does the ordering of the tables make a difference?

2.1.1 Block-Nested Loops Join

1. Optimization of nested loops join.
2. Idea: Rather than scan inner table once per tuple of outer table, scan once per *page* of outer table.

Page accesses count?

3. Further optimization: If we have M buffers, read $M - 2$ pages of outer table then scan inner table.

(1 page each needed for inner table input buffer and output buffer.)

Page accesses count?

2.2 Hash Join

1. Only useful for equijoins.
2. Hash Table 1 on A and Table 2 on B.
3. After hashing, tuples which can possibly join are in the same bucket.
4. Re-read each bucket, separate by table, and join.
5. Assuming each bucket fits into memory, how many page accesses?