CS W186 Introduction to Database Systems
Spring 2020 Josh Hug, Michael Ball

DIS 6

1 Assorted Joins

- Companies: (company_id, industry, ipo_date)
- NYSE: (company_id, date, trade, quantity)

We have 20 pages of memory, and we want to join two tables Companies and NYSE on C.company_id = N.company_id. Attribute company_id is the primary key for Companies. For every tuple in Companies, assume there are 4 matching tuples in NYSE.

NYSE contains [N] = 100 pages, NYSE holds pN = 100 tuples per page.

Companies contains [C] = 50 pages, C holds pC = 50 tuples per page.

There are unclustered B+ indexes on C.company_id and N.company_id – for both indexes, assume it takes 2 I/Os to access a leaf.

(a) How many disk I/Os are needed to perform a simple nested loops join?

[C] + pC*[C]*[N] = 50 + 50*50*100 = 250050

(b) How many disk I/Os are needed to perform a block nested loops join?

(# pages in smaller relation) + ceil[(# pages in smaller relation) / (# pages in memory – 2)] * (# pages in larger relation) = 50 + ceil(50/18) * 100 = 350 I/Os

(c) How many disk I/Os are needed to perform an index nested loops join?

[C] + [C] * pC * (cost to find matching NYSE tuples) = 50 + 50 * 50 * (2 + 4) = 15,050

(d) For this part only, assume the index on NYSE.company_id is clustered. What is the cost of an index nested loops join using companies as the outer relation?

[C] + [C]*pC * (cost to find matching NYSE tuples) = 50 + 50 * 50 * (2 + ceil(4/100)) = 7,550 I/Os

(e) How many disk I/Os are needed to perform a sort merge join? This is ordinary sort-merge, not optimized sort-merge.

Sorting C: 4 * (50 pages) = 200 I/Os
Sorting N: 4 * (100 pages) = 400 I/Os
Joining: [C] + [N] = 150 I/Os
Total: 200 + 400 + 150 = 750 I/Os

(f) How many disk I/Os are needed to perform a hash join?

No recursive partitioning required. Partitioning phase:
ceil([N]/(B - 1))= 6 pages per partition for N, 19(6) pages
ceil([C]/(B - 1)) = 3 pages per partition for C, 19(3) pages
Partitioning phase: 100 + 19(6) + 50 + 19(3) = 321
Build and Probe: 19(6) + 19(3) = 171
Total: 321 + 171 = **492 I/Os**

2 Grace Hash Join

We have 2 tables – Catalog and Transactions.

Catalog has a total of 100 pages and 20 tuples per page. Transactions has a total of 50 pages and 50 tuples per page. Assume that the distribution among the key that we are joining on is uniform for the two tables.

(a) If we had 10 buffer pages, how many partitioning phases would we require for grace hash join? Consider which table we should build the hash table in the probing phase on.

T is smaller, need partitions of T to be at most B - 2 = 8 pages. After 1 partitioning pass, we have partitions of size 6. 6 <= 8 so it's small enough meaning we still only need **1 partitioning pass.**

(b) What is the I/O cost for the grace hash join then?

We need 1 partitioning pass.
Partitioning phase:
ceil([C]/(B - 1)) = 12 pages per partition for C, 12(9) pages in total after partitioning
ceil([T]/(B - 1)) = 6 pages per partition for T, 6(9) pages in total after partitioning
Partitioning IOs: 100 + 12(9) + 50 + 6(9) = 312
Probing phase: 12(9) + 6(9) = 162 I/Os
Total: 312 + 162 = **474 I/Os**

(c) For the above question, if we only had 8 buffer pages, how many number of partition phases would there be?

T is smaller, need partitions of T to be at most B - 2 = 6 pages. After 1 partitioning pass, we have partitions of size 8, which is too big. We need a second partitioning pass. 8 / 7 = 1.1 → 2 pages, which is small enough so we need **2 passes**.

(d) What will be the I/O cost?

Partitioning phase:
ceil([C]/(B - 1)) = 15 pages per partition for C
ceil([T]/(B - 1)) = 8 pages per partition for T
ceil([C]/(B - 1)) = 3 pages per partition for second pass for C
ceil([T]/(B - 1)) = 2 pages per partition for second pass for T
Partitioning IOs: [100 + 50] (1st read) + [15(7) + 8(7)] (first write) + [15(7) + 8(7)] (second

read) + [3(49) + 2(49)] (second write) = 717 I/Os
Build and Probe Phase: 3(49) + 2(49) = 245 IOs
Total: 717 + 245 = **962 I/Os**