**Homework Assignment 2**

**Databases II**

**External Sort**

**Exercise 1.** Consider a relation that is stored in 4000 pages.

1. **Compute the number of passes and the number of I/Os for sorting the relation using the two-way merge sort algorithm. (5 marks)**

The number of passes is [log2N] + 1 , where N is the number of pages in the file,

[log24000] + 1 = 13

The overall cost is 2N (log2N + 1),

2• 4000 (log24000 + 1) = 104 000

Therefore, sorting the relation using the two-way merge sort algorithm takes **13 passes**, and in each pass we read and write 4000 pages, for a total of **103 727 I/Os**.

1. **Compute the number of passes and the number of I/Os for sorting the relation using the external merge sort algorithm with a buffer that consists of 50 pages. (5 marks)**

The number of passes is [log(B-1) [N/B]] + 1 , where N is the number of pages in the file B is the buffer,

[log(50-1) [4000/50]] + 1 = 3

The cost formula is 2 \* N \* (# of passes),

2 \* 4000 \* 3 = 24 000

Therefore, sorting the relation using the external merge sort algorithm takes **3 passes**, and in each pass we read and write 4000 pages, for a total of **24 000 I/Os**.

1. **Recompute the answer to (2) with double buffering. (5 marks)**

Due to double buffering,

B = 50/2 = 25

The number of passes is [log(B-1) [N/B]] + 1 , where N is the number of pages in the file B is the buffer,

[log(25-1) [4000/50]] + 1 = 3

The cost formula is 2 \* N \* (# of passes),

2 \* 4000 \* 3 = 24 000

Therefore, sorting the relation using the external merge sort algorithm takes **3 passes**, and in each pass we read and write 4000 pages, for a total of **24 000 I/Os**.

1. **Recompute the answer to (2) with blocks of size 5. (5 marks)**

With block size of 5, the formula used is [log(B/b-1) [N/B]] + 1 , where b is the block size,

[log (50/5−1) [4000/50]] + 1 = 3

The cost formula is 2 \* N \* (# of passes),

2 \* 4000 \* 3 = 24 000

Therefore, sorting the relation using the external merge sort algorithm with block size of 5 takes **3 passes** and **24 000 I/Os**.

**Query Evaluation**

**Exercise 2.**

1. **Write a SQL query that finds the pairs of sailor names who have reservations on the same day, and translate it to a relational algebra expression. (10 marks)**

SQL Query:

SELECT sname FROM Sailors INNER JOIN Reserves on Reserves.sid = Sailor.sid GROUP BY Reserves.day

Relational Algebra:

𝜋 sname (GROUP BY(Reserves .day) (Reserves)⋈ sid=Sailors sid)

1. **For each item, decide whether the index matches the condition (S = Sailors and R = Reserves). Explain and justify your decision.**

(a) A B+-tree index on the search key hS.sid,S.agei and σS.id=10∧S.age>25(S). (4 marks)

This would match because the condition contains the attributes of the prefix of the B+ tree index search key.

(b) A B+-tree index on the search key hS.age,S.sidi and σS.age>25∧S.day>2020/12/28(S). (4 marks)

There would be no match because the condition contains S.day which is not present in the prefix.

(c) A hash index on the search key hS.age,S.sidi and σS.age=25(S). (4 marks)

There would be no match because the condition does not contain all the attributes in the search key.

(d) Two hash indexes on the search keys S.sid and S.age and σS.age=25∨S.sid=10(S). (4 marks)

There would be no match because all a hash index cannot be used for an OR comparison.

(e) Two B+-tree indexes on the search keys S.sid and hS.age, S.dayi and σS.sid=10∨S.day>2020/12/28(S). (4 marks)

There would be no match because the OR statement does not contain all the prefixes (S.age is missing) in the B+ -tree index.

**3. Consider the following query:**

(a) Estimate the reduction factor of the condition in the query using a B+-tree index I on R.day with High(H) = 2019/01/27 and Low(H) = 2018/12/02. Explain any assumptions about the data distribution in your estimation. (15 marks)

The reduction factor formula is ,

= ≈ 0.43

Assumptions:

-B+ tree index distributes all day values uniformly.

(b) Using the estimation in (a) and assuming that index I is clustered, estimate the number of I/Os to answer the query. Assume that accessing the data entries in index I costs 2 I/Os. (10 marks)

Due to a clustered index, each data entry is pointed to a page of the Reserves relation, so page accesses would be,

0. 43 × 1000 = 429 pages

Then # of I/Os,

429 × 2 = 858 I/Os

Therefore, the estimated number of I/Os to answer the query is about **858 I/Os**.

**Exercise 3.**

**Run the following command and report the result. Specify the join implementation used in the plan, e.g. merge-sort or hash join. Report the estimate of the total cost of running the query. (20 marks)**

cs4411=# explain ( format json ) select \* from counties c , maskuse m where c . fids = m . fids and c.reportdate = ' 2021/01/09 ';

QUERY PLAN

-------------------------------------------------------------

[ +

{ +

"Plan": { +

"Node Type": "Gather", +

"Parallel Aware": false, +

"Startup Cost": 1094.69, +

"Total Cost": 14826.47, +

"Plan Rows": 3358, +

"Plan Width": 73, +

"Workers Planned": 2, +

"Single Copy": false, +

"Plans": [ +

{ +

"Node Type": "Hash Join", +

"Parent Relationship": "Outer", +

"Parallel Aware": false, +

"Join Type": "Inner", +

"Startup Cost": 94.70, +

"Total Cost": 13490.67, +

"Plan Rows": 1399, +

"Plan Width": 73, +

"Inner Unique": false, +

"Hash Cond": "(c.fids = m.fids)", +

"Plans": [ +

{ +

"Node Type": "Seq Scan", +

"Parent Relationship": "Outer", +

"Parallel Aware": true, +

"Relation Name": "counties", +

"Alias": "c", +

"Startup Cost": 0.00, +

"Total Cost": 13376.56, +

"Plan Rows": 1447, +

"Plan Width": 45, +

"Filter": "(reportdate = '2021-01-09'::date)"+

}, +

{ +

"Node Type": "Hash", +

"Parent Relationship": "Inner", +

"Parallel Aware": false, +

"Startup Cost": 55.42, +

"Total Cost": 55.42, +

"Plan Rows": 3142, +

"Plan Width": 28, +

"Plans": [ +

{ +

"Node Type": "Seq Scan", +

"Parent Relationship": "Outer", +

"Parallel Aware": false, +

"Relation Name": "maskuse", +

"Alias": "m", +

"Startup Cost": 0.00, +

"Total Cost": 55.42, +

"Plan Rows": 3142, +

"Plan Width": 28 +

} +

] +

} +

] +

} +

:

The join implementation used in the plan was ‘Hash Join’ and the estimate total cost of running the query was 14826.47.

**Rerun the explain command and report the alternative join algorithm and the new cost estimate. Continue by disabling the alternative algorithm and report the result for the second alternative algorithm. (20 marks)**

cs4411=# set enable\_hashjoin = off;

SET

cs4411=# explain ( format json ) select \* from counties c , maskuse m where c . fids = m . fids and c.reportdate = ' 2021/01/09 ';

QUERY PLAN

-----------------------------------------------------------------

[ +

{ +

"Plan": { +

"Node Type": "Gather", +

"Parallel Aware": false, +

"Startup Cost": 14690.45, +

"Total Cost": 15062.90, +

"Plan Rows": 3358, +

"Plan Width": 73, +

"Workers Planned": 2, +

"Single Copy": false, +

"Plans": [ +

{ +

"Node Type": "Merge Join", +

"Parent Relationship": "Outer", +

"Parallel Aware": false, +

"Join Type": "Inner", +

"Startup Cost": 13690.45, +

"Total Cost": 13727.10, +

"Plan Rows": 1399, +

"Plan Width": 73, +

"Inner Unique": false, +

"Merge Cond": "(c.fids = m.fids)", +

"Plans": [ +

{ +

"Node Type": "Sort", +

"Parent Relationship": "Outer", +

"Parallel Aware": false, +

"Startup Cost": 13452.52, +

"Total Cost": 13456.13, +

"Plan Rows": 1447, +

"Plan Width": 45, +

"Sort Key": ["c.fids"], +

"Plans": [ +

{ +

"Node Type": "Seq Scan", +

"Parent Relationship": "Outer", +

"Parallel Aware": true, +

"Relation Name": "counties", +

"Alias": "c", +

"Startup Cost": 0.00, +

"Total Cost": 13376.56, +

"Plan Rows": 1447, +

"Plan Width": 45, +

"Filter": "(reportdate = '2021-01-09'::date)"+

} +

] +

}, +

{ +

"Node Type": "Sort", +

"Parent Relationship": "Inner", +

"Parallel Aware": false, +

"Startup Cost": 237.93, +

"Total Cost": 245.79, +

"Plan Rows": 3142, +

"Plan Width": 28, +

"Sort Key": ["m.fids"], +

"Plans": [ +

{ +

"Node Type": "Seq Scan", +

:

The join implementation used in the plan was ‘Merge Join’ and the estimate total cost of running the query was 15062.90.

cs4411=# set enable\_mergejoin = off;

SET

cs4411=# explain ( format json ) select \* from counties c , maskuse m where c . fids = m . fids and c.reportdate = ' 2021/01/09 ';

QUERY PLAN

-------------------------------------------------------------

[ +

{ +

"Plan": { +

"Node Type": "Gather", +

"Parallel Aware": false, +

"Startup Cost": 1000.00, +

"Total Cost": 151736.02, +

"Plan Rows": 3358, +

"Plan Width": 73, +

"Workers Planned": 2, +

"Single Copy": false, +

"Plans": [ +

{ +

"Node Type": "Nested Loop", +

"Parent Relationship": "Outer", +

"Parallel Aware": false, +

"Join Type": "Inner", +

"Startup Cost": 0.00, +

"Total Cost": 150400.22, +

"Plan Rows": 1399, +

"Plan Width": 73, +

"Inner Unique": false, +

"Join Filter": "(c.fids = m.fids)", +

"Plans": [ +

{ +

"Node Type": "Seq Scan", +

"Parent Relationship": "Outer", +

"Parallel Aware": true, +

"Relation Name": "counties", +

"Alias": "c", +

"Startup Cost": 0.00, +

"Total Cost": 13376.56, +

"Plan Rows": 1447, +

"Plan Width": 45, +

"Filter": "(reportdate = '2021-01-09'::date)"+

}, +

{ +

"Node Type": "Seq Scan", +

"Parent Relationship": "Inner", +

"Parallel Aware": false, +

"Relation Name": "maskuse", +

"Alias": "m", +

"Startup Cost": 0.00, +

"Total Cost": 55.42, +

"Plan Rows": 3142, +

"Plan Width": 28 +

} +

] +

} +

] +

} +

} +

]

(1 row)

The join implementation used in the plan was ‘Nested Loop’ and the estimate total cost of running the query was 151736.02.