COMP90050 Advanced Database Systems: Tutorial
Winter term, 2023 (Week 2)

Exercises

Part 1:

1. Which of the following RAID configurations that we saw in class has the lowest disk space utilization? Your answer needs to have explanations with calculations for each case.

   (a) RAID 0 with 2 disks

   (b) RAID 1 with 2 disks

   (c) RAID 3 with 3 disks

Where does this lack of utilization of space go, i.e., where we can use such a configuration as it has some benefits gained due to the loss of space utilization?

Solution:

a. In case 1, the space utilization is 100% because the two disks store contiguous blocks of a file in RAID 0.
b. In case 2, the space utilization is 50% because RAID 1 uses mirroring (same data on both disks). MTTF increases, so for cases where a disk can fail easily -- this setting is good. The system operates even when a disk fails.
c. In case 3, the space utilization is (3-1)/3=66.7 because RAID 3 uses one disk for storing parity data.
   Case 2 has the lowest disk space utilization with explanation as given underlined above as a part of the answer.

2. What is the Mean time to failure values of different RAID systems?
   a. RAID 0 with 2 disks
   b. RAID 2 with 2 disks
   c. RAID 1 with 2 disks
   d. RAID 1 with 3 disks
   e. RAID 3 with 3 disks
   f. RAID 4 with 3 disks
   g. RAID 5 with 3 disks
   h. RAID 6 with 5 disks

Let’s say the probability of one disk failure is p, and the MTTF of one individual disk is MTTF.

a+b. RAID 0 with 2 disks and RAID 2 with 2 disks – The system fails is one of the disks fails. The probability that one of the two disks fails (disk A or disk B) is p+p = 2p. So, Mean time to failure of the system is $\frac{1}{2p} = \frac{1}{2} * \frac{1}{p} = \frac{1}{2} * MTTF$

c. RAID 1 with 2 disks – The system fails is both of the disks fail at the same time. The probability that both disks fail (disk A and disk B) is p*p = p^2. So, Mean time to failure of the system is $\frac{1}{p^2} = \left(\frac{1}{P}\right)^2 = MTTF^2$

d. RAID 1 with 3 disks – The system fails is three of the disks fail at the same time. The probability that all disks fail (disk A and disk B and disk C) is p*p*p = p^3. So, Mean time to failure of the system is $\frac{1}{p^3} = \left(\frac{1}{P}\right)^3 = MTTF^3$

e+f+g. RAID 3 with 3 disks and RAID 4 with 3 disks and RAID 5 with 3 disks- The system fails if 2 of the 3 disks fail at the same time. The probability that 2 disks fail is p*p = p^2. There are 3 different possible combinations of 2 disks failures (A,B; or A,C; or B,C disks), so the probability that any of the 2 disks out of these 3 disks fail is $3p^2$. Mean time to failure of the system is $\frac{1}{3p^2} = \left(\frac{1}{3}\right) * \left(\frac{1}{P}\right)^2 = \left(\frac{1}{3}\right) * MTTF^2$

h. RAID 6 with 5 disks - The system fails if 3 of the 5 disks fail at the same time. The probability that 3 disks fail is p*p*p = p^3. There are 10 different possible combinations of 3 disks failures out of 5 disks(A,B,C; or A,B,D; or A,B,E; A,C,D; or A,C,E; or A,D,E; or B,C,D; or B,C,E; or B,D,E; or C,D,E disks), so the probability that any of the 3 disks out of these 5 disks fail is $10p^3$. Mean time to failure of the system is $\frac{1}{10p^3} = \left(\frac{1}{10}\right) * \left(\frac{1}{P}\right)^3 = \left(\frac{1}{10}\right) * MTTF^3$

3. In a Failvote system, which of the following cases we can accept an action?

Total number of devices	Number of agreeing devices	Accept?
10	6	Y
10	5	N
10	4	N
5	3	Y
5	2	N

4. In a Failfast system, which of the following cases we can accept an action?

Total number of devices	Number of working devices	Number of agreeing devices	Accept?
10	6	4	y
10	6	3	n
10	5	3	y
5	5	3	y
5	4	2	n
5	2	2	y
5	1	-	n

5. There are two nodes in a network that use stable storage and acknowledgment message passing for reliable communication. The stable storage of Node A contains the following record - Received message (In6); Transmitted message(Out3); Out:3 Ack:3 In:6. The stable storage of Node B contains the following record - Received message (In3); Transmitted message(Out6); Out:6 Ack:6 In:3.

Now Node B sends a new message 7 to Node A. What will be in the stable storage of A and B if the message is received correctly, including a correctly received acknowledgement?

> Node A: Received message (In**7**); Transmitted message(Out3); Out:3 Ack:3 In:**7**.
> Node B: Received message (In3); Transmitted message(Out**7**); Out:**7** Ack:**7** In:3.

6. There are two nodes in a network that use stable storage and acknowledgment message passing for reliable communication. The stable storage of Node A contains the following record - Received message (In6); Transmitted message(Out3); Out:3 Ack:3 In:6. The stable storage of Node B contains the following record - Received message (In3); Transmitted message(Out6); Out:6 Ack:6 In:3.

Now Node B sends a new message 7 to Node A. **What will change in the stable storage of A and B if the message is received correctly, but the acknowledgement is not received?**

> Node A: Received message (In**7**); Transmitted message(Out3); Out:3 Ack:3 In:**7**.
> Node B: Received message (In3); Transmitted message(Out**7**); Out:**7** Ack:**6** In:3.

Part 2:

7. Discuss which of the query optimisation approach(es) can be used/more suitable for the following scenarios: a. enumerating all plans; b. heuristic based; c. adaptive plans -

- Scenario A: Given a table with 1000 tuples, run the following query:

```
SELECT customer
FROM Table
WHERE spend BETWEEN 100 AND 200
AND birth_year> 2000;
```

- Scenario B: Given 5 tables with 1000 tuples in each table, run a query:

```
SELECT T1.name, T2.salary, T3.qualification, T4.phone, T5.leader
FROM Table1 T1
   INNER JOIN Table2 T2 ON T2.id = T1.id
   INNER JOIN Table3 T3 ON T3.id = T1.id
   INNER JOIN Table4 T4 ON T4.id = T1.id
   INNER JOIN Table5 T5 ON T5.department = T1.department
WHERE T1.age > 50;
```

Solution:

Queries are generally converted to Relational Algebra expressions internally first. Then the system tries to create alternate plans and pick the best plan to execute in terms of execution time. There are two general approaches for this. One is searching/enumerating all the plans and choose the best one. Another approach is using heuristics to choose a plan (or a combination of two can be done as well).

Adaptive plan is executing a part/some parts of a query plan first to re-evaluate the cost of the other parts of the query plan that haven't been executed yet, to have a better overall cost estimation (and hence, choosing a better plan).

For Scenario A, the heuristic approach can be suitable due to the simplicity of the query and small size of the table. For Scenario B, the enumerating approach can be more suitable due to the complexity of the query.

Adaptive plan is used for better estimation of cost, hence, cannot be used when the query optimiser is purely heuristic based for a query (so cannot be used for scenario A if the plan is heuristic based). Adaptive plan can be used for cost-based query optimiser (either exhaustive enumeration of all plans or a combination), hence, can be used for Scenario B.

8. Review the examples on nested-loop join and block nested-loop join given in the lecture. Discuss and calculate why the later one can be more efficient.

Solution:
After reiterating the examples/calculations, we see the later one is most efficient because each block in the inner relation is read once for each block in the outer relation (instead of once for each tuple in the outer relation).

9. A particular query on table A used to run quite efficiently in a DBMS. After inserting many records and deleting many other records from table A, that same query is now taking more time to run, even when the total number of records has not changed. What can be the reason for that? What can you do as the user/database administrator of that DBMS to improve the performance of this query?

Solution:
After insertions and deletions, the statistics of a table may not be updated instantly. As the statistics are used to estimate the cost of a query, wrong statistics are causing wrong cost estimations, and hence the query optimiser is now choosing a query plan that is no longer optimal for that query.

Enforcing statistical recompilation option can be used to update the statistics of the table.

Part 3:

10. What indices are more suitable if a table is frequently used for finding records based on the following criteria: users' name, a range of users' birthday, and a spatial region covering users' residence?

    Solution:

    • Users' name: Hash index.
    • Users' birthday: B+ tree index.
    • Users' residence: R-tree index or another spatial index such as a quadtree index.

11. Review the points on indexing with B+ trees. Assume a database table has 10,000,000 records and the index is built with a B+ tree. The maximum number of children of a node, is denoted as n. How many steps are needed to find a record if n=4? How many steps are needed to find a record if n=100?

Solution:

• When n = 4, the maximum height of the tree is

$\lceil \log_{\lceil n/2 \rceil}(K) \rceil = \lceil \log_2(10000000) \rceil = 24$

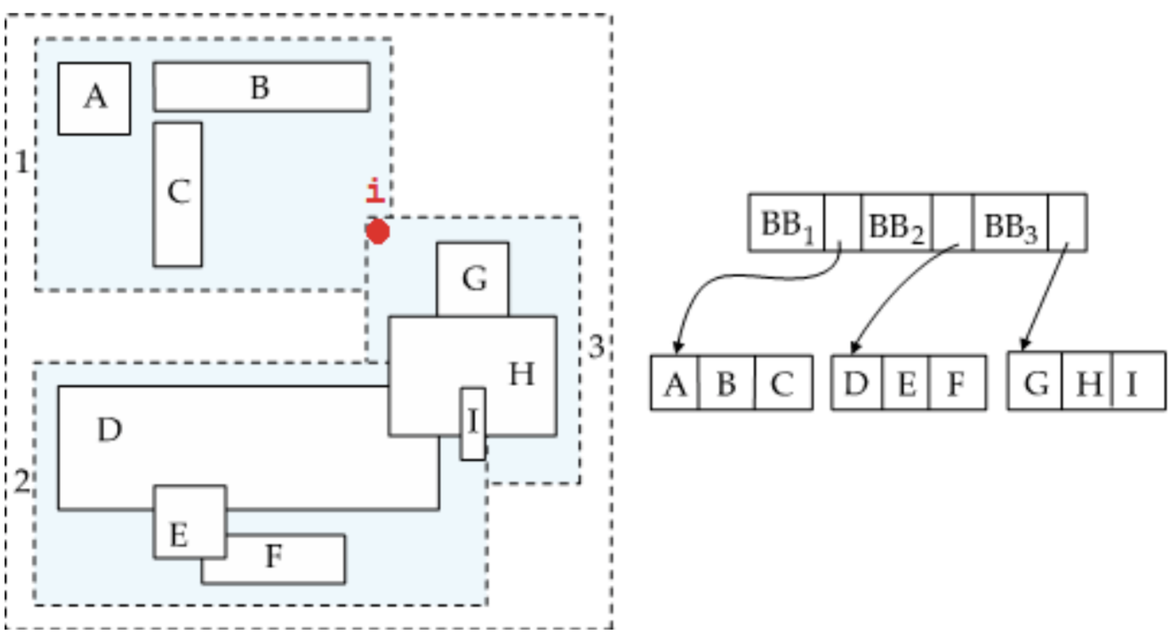
Therefore, 24 steps are needed.

• When n = 100, the maximum height of the tree is

$\lceil \log_{\lceil n/2 \rceil}(K) \rceil = \lceil \log_{50}(10000000) \rceil = 5$

Therefore, 5 steps are needed.

12. Given the R-tree below please visit the nodes of the R-tree in a best-first manner as discussed in class to find the 1st nearest neighbour of query point "i". Is there anything peculiar that you notice while traversing an R-tree?



Solution:

In this traversal we first visit node BB1 as it overlaps with the query point. And find that object B is the closest to query point i. The issue here is that we cannot stop at this point in the traversal as i overlaps with BB3 as well so we need to investigate the data there too. We then figure out that G is the closest object overall. Due to overlaps in R-tree branches two or more branches of an R-tree need to be investigated in many query types. In addition, as each internal node represents a bounding box, thus we are not sure about the position of objects in a bounding box which may necessitate that we investigate multiple bounding boxes to determine a nearest neighbour in this case.