Advanced Database Management Systems

Date: Tuesday 22nd January 2013
Time: 09:45 - 11:45

Please answer BOTH Questions

The use of the word BRIEFLY indicates that you answers should not be more than 5-7 sentences long.

Marks will be awarded for reasoning and method as well as being correct.

The total examination mark is out of a possible 60.

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text.
1. a) Briefly explain the three common routes by means of which users and/or applications can issue queries and transactions against a database. (3 marks)

b) Briefly explain the difference between the rewriting stage and the plan selection stage inside a query optimiser. (2 marks)

c) Consider the following relation schemas:

   suppliers (sid: integer, sname: string, address: string)
   parts (pid: integer, pname: string, colour: string)
   catalogue (sid: integer, pid: integer, cost: real)

   i) Write the following query in SQL: “Return the address of suppliers who supply green parts costing less than 5 pounds”. (3 marks)

   ii) Draw a logical operator tree that would classically result from the direct, clause-by-clause translation into relational algebra of the SQL query you gave as your answer to the preceding item. (3 marks)

   iii) Taking the operator tree you gave as your answer to the preceding item as your starting point, draw the logical operator tree that would classically result from an equivalence-based rewriting process whose intent was to push selections and introduce projections as close to the leaves as possible, as well as replacing with a join any Cartesian product whose parent is a selection on its attributes. (12 marks)

iv) Consider the following SQL query over the schemas above:

   ```sql
   select *
   from  suppliers s, catalogue
   where s.sid = c.sid
   ```

   Assume that this query is to run over three shared-nothing computational nodes, call them N1, N2 and N3. Assume further that the extents of both suppliers and parts are hash-partitioned between N1 and N2, i.e., a certain portion of the extent of suppliers is stored in N1 and the remainder in N2, and the same applies for the extent of parts.

   Draw an operator tree for the query in which the equijoin is run in N1 and N2, i.e., with a degree of partitioned parallelism equal to 2, with the partial results converging to N3 for merging into the final result.

   In your answer, focus on ensuring that you capture clearly (a) the computational nodes and their stores; and (b) each copy of the scan, join, split and merge operators paying particular attention to the dataflow edges they must form. (7 marks)
2. a) Consider again the following relation schemas (already shown above):

```
suppliers (sid: integer, sname: string, address: string)
parts (pid: integer, pname: string, colour: string)
catalogue (sid: integer, pid: integer, cost: real)
```

Now, assume that the estimated size of `suppliers` is twice the estimated size of `parts` but only half the estimated size of `catalogue`. Finally, assume that the database that stores them is distributed as indicated by the following join graph:

```
i) Using the notation described in the course, state, step by step, the evaluation program that is heuristically most efficient, given the information above, for the following query injected at Site 3:

```
select c.cost
from suppliers s, parts p, catalogue c
where s.sid = c. sid and p.pid = c.cid
```

(6 marks)

ii) Briefly explain why an evaluation program that scheduled all the operators in the query for execution at Site 2 is unlikely, heuristically speaking, to be more efficient under classical assumptions. (2 marks)

iii) Consider again the query in the previous item and the problem of selecting an evaluation program. Briefly explain why semijoins are often useful in distributed query evaluation, what the potential downside is and whether or not that downside applies in the case of this query. (6 marks)

b) Briefly explain what common property of the join, sort and duplicate removal operations can be used to characterise a striking difference between query evaluation over stored extents and streams. (1 mark)

c) Briefly explain why map-reduce computations so directly capture join-free group-by queries. (1 mark)

d) State two of the four characteristics of so-called big data known as the four Vs. (2 marks)

e) Briefly explain the difference between scale-up and scale-out and why it is said that with big data scale-up is not an option and only scale-out works. (2 marks)
f) Consider the routing tree below. Arrows denote child-to-parent communication links, nodes are labelled with the hop count to reach the base station, and a list associated with an edge denotes the transmission load in that edge, as follows: the length of the list denotes the number of messages going through that edge whereas the number of bytes transmitted is the sum of the length (in characters) of each element in the list.

Draw an equally-shaped routing tree but labelling the edges under the assumption that one is doing a tree-staged \textit{COUNT} aggregation, rather than sending all the data back to the base station as in the routing tree above, then calculate the reduction obtained in terms of number of messages and number of bytes by using tree-staged aggregation.

(9 marks)

g) State whether TinyDB, or SNEE, or both, use tree-staged aggregation.

(1 mark)