CS632 Midsem, Spring 2016, Feb 24, 2016, 8.30-10.30 AM

Total Marks: 50

NOTE: Answer all subparts of a question together, do not split them up.

1. Volcano and Multi query optimization

   (a) In the context of Volcano, if the physical property sort on A is required, applying an enforcer will result in the input requiring no (null) physical property. However, for multi-query optimization, we may wish to generate sort on A by sorting a result that is already sorted on B, and vice versa. Thus, with MQO there can be cycles within the physical equivalence nodes corresponding to a single logical equivalence node, which cannot occur with basic Volcano. Why do we need to allow such cycles for multiquery optimization? Explain briefly. ...3

   (b) An MQO algorithm must find the best plan for each materialized node. Note the equations for finding the cost of a physical equivalence node and a physical operator node from the paper. If two physical equivalence nodes within a logical equivalence node are materialized, explain how the equations can result in cyclic plans that are impossible to execute. ...4

   NOTE: The MQO paper you read does not address the above problem due to lack of space. However a solution based on spanning trees is given in Prasan Roy's thesis.

   (c) Consider an equivalence rule such as $r = \Pi_R(r \bowtie s)$ with schemas $r(A, B), s(B, C)$, where $r.B$ is a foreign key to $s.B$, and is not nullable. Such a rule can be used in a subsumption derivation, if $r \bowtie s$ is materialized, to compute r from $r \bowtie s$ (which could be useful if r is a complex expression, not a relation). Explain why having such rules would cause problems for the optimizer. ...4

2. Query Optimization Misc.

   (a) What is the need for the max1row operator? And give two different sufficient conditions under which it can be omitted, using simple example queries. ...1+3

   (b) Consider the plan bouquet paper. Using the query $r \bowtie_{p1} \sigma_{p2}(s)$. where both predicates $p1$ and $p2$ are error prone, explain how lack of independence between predicates $p1$ and $p2$ can affect the cost guarantees provided by the paper. ...4

3. SCOPE

   (a) The SCOPE paper mentions a few rules for inferring functional dependencies and column equivalence classes. (a) If two columns $r.A$ and $s.B$ are in the same equivalence class, what functional dependencies can you infer between them? (b) Give a rule for inferring a (non-trivial) functional dependency on a join result (other than those derived from column equivalence classes). ...1+4

   (b) Explain why partitioning on $\{C1, C2\}^g$ does not imply partitioning on $\{C1\}^g$ using an example. Also give a proof sketch for the converse, i.e. $\{C1\}^g \Rightarrow \{C1, C2\}^g$. ...4

4. Main-memory databases: In the context of NUMA multicore join evaluation, An alternative to having a shared hash table (with probe relation partitioned) is to partition both the relations between the cores, and then perform join locally. What are the benefits and drawbacks of such an approach as compared to having a shared hash table used in the Morsel driven parallelism paper? ...3

5. Calvin

(a) Checkpoints in Calvin need to be transaction consistent, whereas with ARIES, checkpoints need not be transaction consistent. Explain Calvin needs transaction consistent checkpoints. ...3

(b) Recovery in Calvin makes a hidden assumption related to remote reads when replaying logs, without giving any explanation of how to fix it. In other words, as stated recovery in Calvin is buggy. Explain the bug. Also give one possible solution. ...3+3

6. PNUTS: Version numbering as described in the PNUTS paper is based on a per item counter. Suppose instead that it is based on a timestamp, which can be assumed to change fast enough to be unique. This could then potentially be used to ensure that a version at a site is not too "stale", i.e. if there is a new version created at timestamp $t1$, a site that asks for freshness of δ will not get an older version after time $t1 + \delta$.

   Explain how you could implement this scheme efficiently (without always going to the master for the latest version). Note that each storage server can be assumed to have its own clock, which may not be quite in sync with others, but you can assume that updates from a particular storage server will be received in increasing order of timestamp. ...10

Total Marks = 50