CSE 444 final review

Some possible topics

Database tuning
- B+ trees
- Index selection

Query execution and optimization
- Relational algebra
- Physical query operators
- Estimating statistics
- Selecting logical and physical plans

Parallel databases
- Parallel, distributed query operators
- MapReduce
- Pig Latin

Note: this is not a complete list of topics

Other recommended problems to review
- Fall 2009 Final Problem 2
- Fall 2009 Final Problem 3
- * and many more available in previous exams *
- http://www.cs.washington.edu/education/courses/cse444/11wi/exams/index.html

Map Reduce Example

Recall the following pseudo code for counting words in a document.

```
map(String key, String value):
        // key: document name
        // value: document contents
        for each word w in value:
                Emit Intermediate(w, "1"):

reduce(String key, Iterator values):
        // key: a word
        // values: a list of counts
        int result = 0;
        for each v in values:
                result += ParseInt(v);
        Emit(AsString(result));
```

Working with R(<u>A</u>, B) S(<u>C</u>,D), implement the following in Map Reduce pseudo code:

a) Select * from R where B<9   [selection]

b) Select distinct A from R [duplicate elimination]

c) Select * from R, S where R.A = S.C [join]

Cost of query execution example

(a) **[8 points]** Consider two tables $R(A, B)$ and $S(C, D)$ with the following statistics:

$$
\begin{aligned}
B(R) &= 5 \\
T(R) &= 200 \\
V(R, A) &= 10 \\
B(S) &= 100 \\
T(S) &= 400 \\
V(S, C) &= 50 \\
M &= 1000
\end{aligned}
$$

There is a clustered index on $S.C$ and an unclustered index on $R.A$. Consider the logical plan:

$$P = \sigma_{A=??}(R) \bowtie_{B=C} S$$

There are two logical operators, $S = \sigma_{A=??}$ and $J = \bowtie_{B=C}$, and for each we consider two physical operators:

$$
\begin{aligned}
s1 &= \text{one pass table scan} \\
s2 &= \text{index-based selection} \\
j1 &= \text{main memory hash join} \\
j2 &= \text{index-based join}
\end{aligned}
$$

Both $s1$ and $s2$ are pipelined, i.e. the result of the select operator is not materialized. For each of the resulting four physical plans compute its cost in terms number of disc I/0s, expressed as a function of the statistics above. Your answer should consists of four expressions, e.g. $\texttt{COST}(s1j1) = B(R)B(S)/M + V(R, A)$ (not the real answer).

i. $\texttt{COST}(s1j1) =$

ii. $\texttt{COST}(s2j1) =$

iii. $\texttt{COST}(s1j2) =$

iv. $\texttt{COST}(s2j2) =$

(b) **[2 points]** Indicate the cheapest plan of the four, together with its cost expressed as a number. You will get credit for this point only if you compute correctly all four expressions above.

3. (**20** points) **Query Optimization**

Consider the schema R(a,b), S(b,c), T(b,d), U(b,e).

(a) (**5** points) For the following SQL query, give two equivalent logical plans in relational algebra such that one is likely to be more efficient than the other. Indicate which one is likely to be more efficient. Explain.

```
SELECT   R.a
FROM     R, S
WHERE    R.b = S.b
         AND S.c = 3
```

(b) (**5** points) Recall that a *left-deep* plan is typically favored by optimizers. Write a left-deep plan for the following SQL query. You may either draw the plan as a tree or give the relational algebra expression. If you use relational algebra, be sure to use parentheses to indicate the order that the joins should be performed.

```
SELECT   *
FROM     R, S, T, U
WHERE    R.b = S.b
         AND S.b = T.b
         AND T.b = U.b
```

(c) (**3** points) Physical plans. Assume that all tables are clustered on the attribute b, and there are no secondary indexes. All tables are large. Do not assume that any of the relations fit in memory.

For the left-deep plan you gave in (b), suggest an efficient physical plan.

Specify the physical join operators used (hash, nested loop, sortmerge, etc.) and the access methods used to read the tables (sequential scan, index, etc.). Explain why your plan is efficient. For operations where it matters, be sure to include the details — for instance, for a hash join, which relation would be stored in the hash tables; for a loop join, which relation would be the inner or outer loop. You should specify how the topmost join reads the result of the lower one.

(d) (**2** points) For the physical plan you wrote for (c), give the estimated cost in terms of $B(...)$, $V(...)$, and $T(...)$. Explain each term in your expression.

(e) (**3** points) For the same logical plan you derived in (b), suggest yet another physical plan under the following assumptions.

Recall that a *star schema* consists of one large *fact table* and many (relatively) small *dimension tables*. Assume that R is a large fact table clustered on a and S, T, and U are small dimension tables that are unclustered and fit entirely in memory (all at once). Further, assume a secondary index exists on $U.e$.

Specify the physical join operators used (hash, nested loop, sortmerge, etc.) and the access methods used to read the tables (sequential scan, index, etc.)

For operations where it matters, be sure to include the details — for instance, for a hash join, which relation would be stored in the hash tables; for a loop join, which relation would be the inner or outer loop. You should specify how the topmost join reads the result of the lower one.

(f) (**2** points) For the physical plan you wrote for (e), give the estimated cost of your physical plan in terms of $B(...)$, $V(...)$, and $T(...)$. Explain each term in your expression.