SECTION 9

Query Optimization and Pig Latin
Today’s Overview

- Reminders
  - Project 4 due last day of class at 5pm

- Query Optimization
  - 2 examples from previous finals (you can also find these in the relational algebra/query plan worksheet from section 7)

- Pig Latin / Map Reduce
  - 1 example from previous final

- Project 4

- Quiz section evaluations (last ~20 minutes of class)
Warm-up

• Consider the query $R(A,B)$ join $S(C,D)$ join $T(E,F)$
  • the join condition is $B=C$ and $D=E$ and:
    • $M = 100$
    • $B(R) = 30$
    • $B(S) = 200$
    • $B(T) = 60$
    • $B(R \text{ join } S) = 80$
    • $B(S \text{ join } T) = 50$.

• Design an optimal query plan that uses only main-memory hash join algorithms. Your plan may store intermediate results to disk if necessary
Load R & T into memory and create hash tables of them. Then read blocks of S one at a time, performing the joins in the following graph.
Summer 2009 Problem 2/3

• $R(a,b,c)$ and $S(x, y, z)$

<table>
<thead>
<tr>
<th>$B(R) = 600$</th>
<th>$B(S) = 800$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T(R) = 3000$</td>
<td>$T(S) = 4000$</td>
</tr>
<tr>
<td>$V(R, a) = 300$</td>
<td>$V(S, x) = 100$</td>
</tr>
<tr>
<td>$V(R, b) = 100$</td>
<td>$V(S, y) = 400$</td>
</tr>
<tr>
<td>$V(R, c) = 50$</td>
<td>$V(S, z) = 40$</td>
</tr>
</tbody>
</table>

We also have $M = 1000$ (number of memory blocks).

*note this problem is also on the worksheet form section 7*
Summer 2009 Problem 2/3

• Write a SQL query for the plan shown on the previous slide

• Optimize the query plan shown on the previous slide, and tell how much you expect the change to improve the performance

• Specify good physical plan for joining R and S on R.a = S.z. Give estimated cost of solution in terms of # of disk I/Os
raw = LOAD 's3n://uw - cse444 - proj4/excite.log.bz2' USING PigStorage('\t') AS (user, time, query);

a = GROUP raw BY user;
b = FOREACH a GENERATE group AS user, COUNT(raw) AS n_searches;
c = GROUP b ALL;
d = FOREACH c GENERATE AVG(b.n_searches), MIN(b.n_searches), MAX(b.n_searches);

STORE d INTO '/user/hadoop/answer.txt' USING PigStorage();
Questions (a)

• What does this program compute and store into the final output file? Describe what the result is, not the details of how it is computed. (Hint: “GROUP b ALL” sends all tuples of bag/relation b to a single group.)
Answer (a)

- The output contains a single row:
  - avg  min  max

- where
  - avg = mean number of searches by one user
  - min = minimum number of searches by one user
  - max = maximum number of searches by one user

- The actual output when this was run against the excite.log.bz2 data was:
  - 4.705126673040153 1 452
Questions (b)

• In order to run this program, the Pig system translates it into a sequence of one or more Hadoop Map - Reduce jobs.

• Describe the Map - Reduce job(s) needed to execute this program. For each job describe the input and output of each map and reduce phase, including the keys and values at each step.

• You do not need to guess exactly how Pig would translate the program as long as your answer gives a reasonable implementation as a sequence of map - reduce jobs.
Answer (b)

• This program generates two map-reduce jobs.

• Job 1:
  
  • **MAP**
  
  • map 1 input: keys = tuple IDs, values = (user, time, query) tuples.
  
  • map 1 output: key = user, value = •. The actual value doesn’t matter, it could be the integer 1, a single character, or any other marker to record one search by that user.

  • **REDUCE**
  
  • reduce 1 input: key = user, value = [•], i.e., array of search markers.
  
  • reduce 1 output: key = user, value = length of array, i.e., search count for that user.
Answer (b)

• Job 2:
  • **MAP**
    • map 2 input: key = user, value = search count for that user
    • map 2 output: key = “x”, value = search count v. Here the key value doesn’t matter except that it needs to be the same for all map outputs.
  • **REDUCE**
    • reduce 2 input: key = “x” (same as map 2 output key), value = [v], i.e., array of all individual user search counts.
    • reduce 2 output: average, min, and max of values in input array [v].
Pig-Latin References

• Massively Parallel Data Analysis with MapReduce
  • http://www.systems.ethz.ch/education/past-courses/hs08/map-reduce/slides/pig.pdf

• Intro to Pig
  • http://www.cloudera.com/wp-content/uploads/2010/01/IntroToPig.pdf