CSE 444: Database Internals

Section 8: Parallel Processing

Review in this section

- ➢ Parallel DBMS
- ➢ MapReduce

1a. Parallel DBMS

R(a,b) is <u>horizontally partitioned</u> across N = 3 machines.

Each machine locally stores approximately 1/N of the tuples in R.

The tuples are randomly organized across machines (i.e., R is <u>block</u> <u>partitioned</u> across machines).

Show a RA plan for this query and how it will be executed across the N = 3 machines.

Pick an efficient plan that leverages the parallelism as much as possible.

SELECT a, max(b) as topb FROM R WHERE a > 0 GROUP BY a R(a, b)

SELECT a, max(b) as topb FROM R WHERE a > 0 GROUP BY a



SELECT a, max(b) as topb FROM R WHERE a > 0 GROUP BY a

If more than one relation on a machine, then "scan S", "scan R" etc



R(a, b)





R(a, b)

SELECT a, max(b) as topb FROM R WHERE a > 0 GROUP BY a









1b. Map Reduce

Explain how the query will be executed in MapReduce

SELECT a, max(b) as topb FROM R WHERE a > 0 GROUP BY a

Specify the computation performed in the map and the reduce functions

Мар

SELECT a, max(b) as topb FROM R WHERE a > 0 GROUP BY a

- Each map task
 - Scans a block of R
 - Calls the map function for each tuple
 - The map function applies the selection predicate to the tuple
 - For each tuple satisfying the selection, it outputs a record with key = a and value = b

•When each map task scans multiple relations, it needs to output something like **key = a and value = ('R', b)** which has the relation name 'R'

Shuffle

SELECT a, max(b) as topb FROM R WHERE a > 0 GROUP BY a

 The MapReduce engine reshuffles the output of the map phase and groups it on the intermediate key, i.e. the attribute a

•Note that the programmer has to write only the map and reduce functions, the shuffle phase is done by the MapReduce engine (although the programmer can rewrite the partition function), but you should still mention this in HW6 answers.

Reduce

SELECT a, max(b) as topb FROM R WHERE a > 0 GROUP BY a

- Each reduce task
 - computes the aggregate value max(b) = topb for each group (i.e. a) assigned to it (by calling the reduce function)
 - outputs the final results: (a, topb)

A local combiner can be used to compute local max before data gets reshuffled (in the map tasks)

- Multiple aggregates can be output by the reduce phase like
 key = a and value = (sum(b), min(b)) etc.
- Sometimes a second (third etc) level of Map-Reduce phase might be needed

1c. Benefit of hash-partitioning

SELECT a, max(b) as topb

- What would change if we hash-partitioned R on R.a before executing this query
 - For parallel DBMS
 - For MapReduce



1c. Benefit of hash-partitioning

SELECT a, max(b) as topb

- For parallel DBMS
 - It would avoid the data re-shuffling phase
 - It would compute the aggregates locally



1c. Benefit of hash-partitioning WHERE a > 0 GROUP BY a

SELECT a, max(b) as topb

- For MapReduce
 - Logically, MR won't know that the data is hashpartitioned
 - MR treats map and reduce functions as black-boxes and does not perform any optimizations on them
- But, if a local combiner is used
 - Saves communication cost:
 - fewer tuples will be emitted by the map tasks
 - Saves computation cost in the reducers:
 - the reducers would not have to do anything