CSE 444 Practice Problems

Parallel DBMSs and MapReduce

1. **Parallel Data Processing Algorithms**

   (a) Describe how to compute the equi-join of two relations R and S in parallel. Describe how to compute a group by aggregation operation in parallel.

   Solution:
   See lecture notes.

2. **System Comparison**

(a) In a parallel DBMS, why is it difficult to achieve linear speedup and linear scaleup?

Solution:

There are three key reasons why linear speedup and linear scaleup are difficult to achieve:

i. Startup cost: The latency involved in starting an operation on many nodes may dominate the computation time.

ii. Interference: Each new process competes for shared resources with the other processes (e.g., network bandwidth). This resource contention can limit the performance gains of adding more processes.

iii. Skew: The time to complete a job is the time that the slowest partition takes to complete its job. When the variance dominates the mean, increased parallelism improves elapsed time only slightly.

(b) List two features common to a traditional DBMS and MapReduce.

Clarification: Here, "traditional DBMS" means a traditional *parallel DBMS*.

Solution:

Many answers are possible including:

i. Horizontal data and operator partitioning.

ii. Distribution independence: applications need not know that the data is distributed.

(c) List two features that are different between the two types of systems: i.e., features that are present in one system but not in the other. For example, you can give one feature present in MapReduce but absent in a parallel DBMS and one feature present in a parallel DBMS but missing from MapReduce (or any other combination).

Solution:

Again, different answers were possible including:

i. A DBMS offers updates, transactions, indexing, and pipelined parallelism.

ii. MapReduce has intra-query fault-tolerance and handles stragglers better.