CS W186 - Spring 2024
Exam Prep Section 11
Parallel Query Processing

1 Types of Parallelism

For each of the following scenarios, state whether it is an example of:

- Inter-query parallelism
- Intra-query, inter-operator parallelism
- Intra-query, intra-operator parallelism
- No parallelism

1. A query with a selection, followed by a projection, followed by a join, runs on a single machine with one thread.

2. Same as before, but there is a second machine and a second query, running independently of the first machine and the first query.

3. A query with a selection, followed by a projection, runs on a single machine with multiple threads; one thread is given to the selection and one thread is given to the projection.

4. We have a single machine, and it runs recursive hash partitioning (for external hashing) with one thread.

5. We have a multi-machine database, and we are running a join over it. For the join, we are running parallel sort-merge join.

2 Partitioning for Parallelism

1. Suppose we have a table of size 50,000 KB, and our database has 10 machines. Each machine has 100 pages of buffer, and a page is 4 KB.

   We would like to perform parallel sorting on this table, so first, we perfectly range partition the data. Then on each machine, we run standard external sorting.

   How many passes does this external sort on each machine take?

2. Suppose we were doing parallel hash join. The first step is to partition the data across the machines, and we usually use hash partitioning to do this.

   Would range partitioning also work? What about round-robin partitioning?

3. Suppose we have a table of 1200 rows, perfectly range-partitioned across 3 machines in order. We just bought a 4th machine for our database, and we want to run parallel sorting using all 4 machines.

   The first step in parallel sorting is to repartition the data across all 4 machines, using range partitioning. (The new machine will get the last range.)

   For each of the first 3 machines, how many rows will it send across the network during the repartitioning? (You can assume the new ranges are also perfectly uniform.)

3 Parallel Query Processing

Suppose we have 4 machines, each with 10 buffer pages. Machine 1 has a Students table which consists of 100 pages. Each page is 1 KB, and it takes 1 second to send 1 KB of data across the network to another machine.

1. How long would it take to send the data over the network after we uniformly range partition the 100 pages? Assume that we can send data to multiple machines at the same time.

2. Next, imagine that there is another table, Classes, which is 10 pages. Using just one machine, how long would a BNLJ take if each disk access (read or write) takes 0.5 seconds?

3. Now assume that the Students table has already been uniformly range partitioned across the four machines, but Classes is only on Machine 1. How long would a broadcast join take if we perform BNLJ on each machine? Do not worry about the cost of combining the output of the machines.

4. Which algorithm performs better?

5. Knowing that the Students table was range partitioned, how can we improve the performance of the join even further?