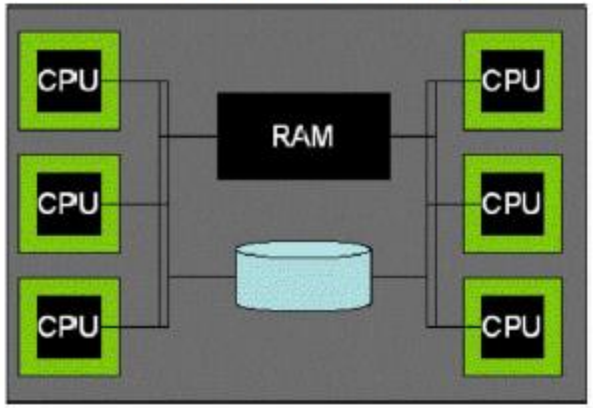
Discussion 11

Parallel Querying

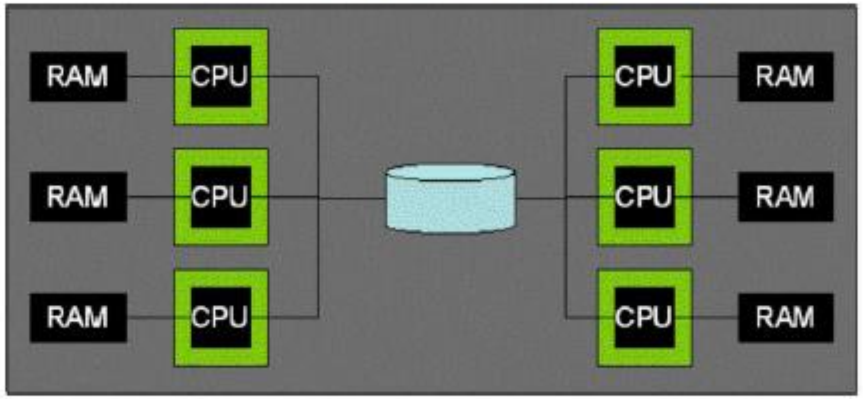
Jiahui Xu

Parallel Architectures

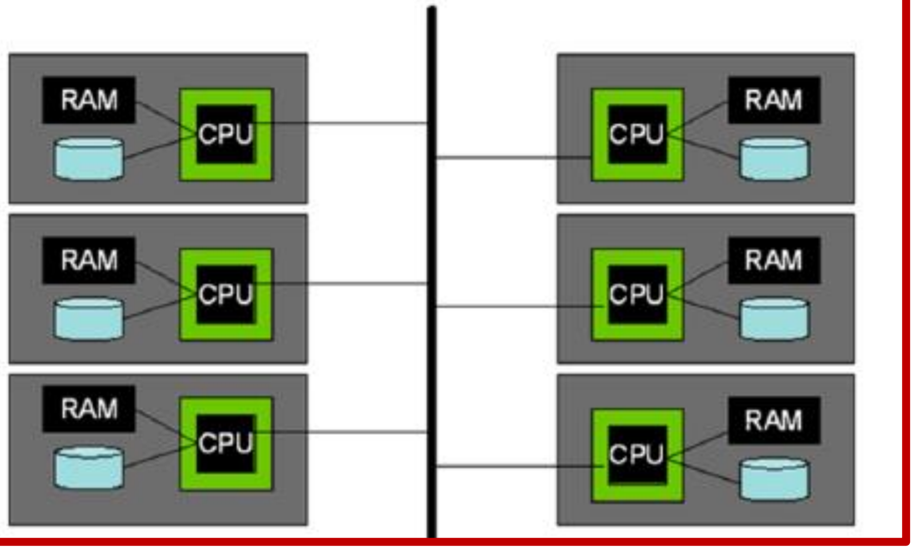
Shared Memory (& Disk)



Shared Disk



Shared Nothing
(cluster)



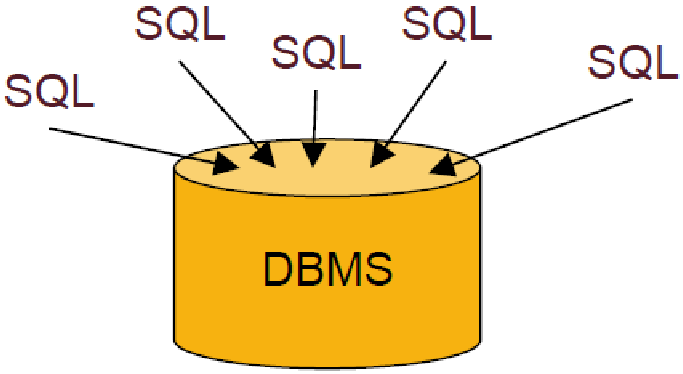
- Do not need to wait for the resource to become available
- The machines communicate with each other solely through the network by passing messages to each other.

Kinds of Query Parallelism

- Parallelism requires **multiple threads/ machine**
- Inter-xx:
    - Different xx
    - Run different xx in parallel
- Intra-xx:
    - One xx
    - Make one xx run as fast as possible

Kinds of Query Parallelism

- Inter-query:
  - Different queries
  - gives **each machine** different queries to work on so that the system can achieve a high throughput and complete as many queries as possible



- Intra-query:
  - One query
  - make one query run as fast as possible by spreading the work over **multiple threads/machines**

Intra-query

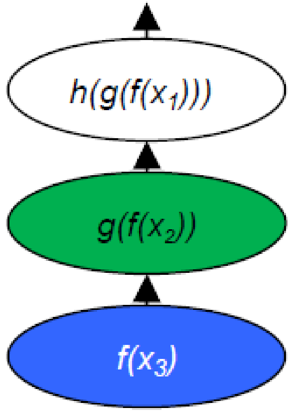
- Intra-query: make one query run as fast as possible by ?

  - Inter operator
    - by running the operators in parallel
    - Pipeline Parallelism
    - Bushy (Tree) Parallelism

  - Intra operator
    - by making one operator run as quickly as possible
    - Partition Parallelism

Intra-query Inter operator

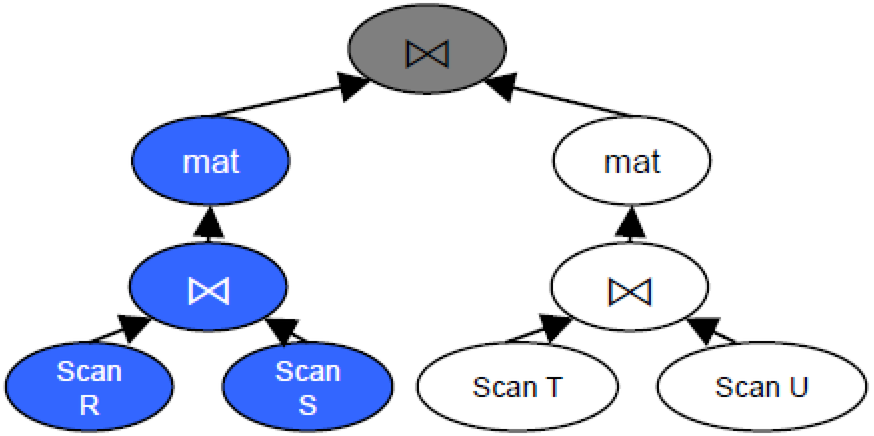
Make one query run as fast as possible by running the operators in parallel



Pipeline Parallelism

- **Parent & child** work at the same time
- Parent can work on a record that its child has already processed while the child is working on next record

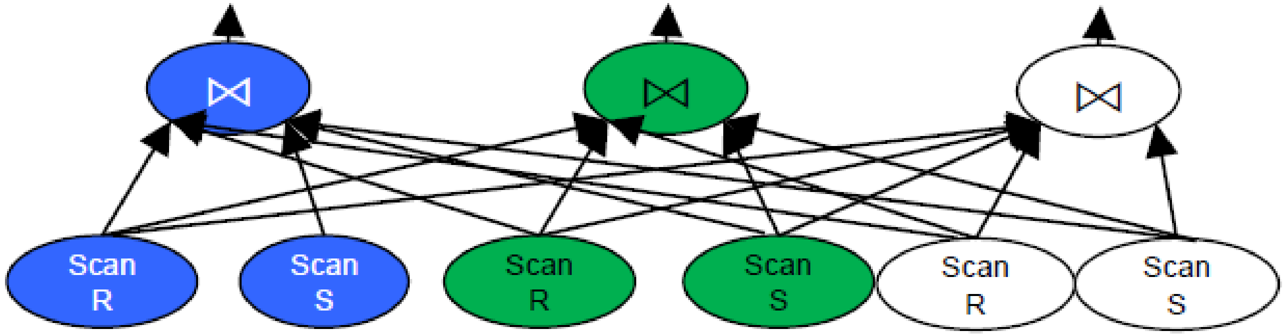
- **Different branches of the tree** are run in parallel



Bushy (Tree) Parallelism

Intra-query Intra operator

Make one query run as fast as possible by making one operator run as quickly as possible



Partition Parallelism

- Dividing up the data onto several machines
- Having them make one operation on the data in parallel

Worksheet

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

Is the query an example of:

1. Inter-query parallelism,
2. Intra-query, Inter-operator parallelism
3. Intra-query, Intra-operator parallelism
4. No parallelism

Worksheet

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

Is the query an example of:

1. Inter-query parallelism,
2. Intra-query, Inter-operator parallelism
3. Intra-query, Intra-operator parallelism
4. No parallelism

Worksheet

A query with a selection, followed by a projection, followed by a join, runs on a single machine with one thread. And there is a second machine and a second query, running independently of the first machine and the first query.

Is the query an example of:

1. Inter-query parallelism,
2. Intra-query, Inter-operator parallelism
3. Intra-query, Intra-operator parallelism
4. No parallelism

Worksheet

A query with a selection, followed by a projection, followed by a join, runs on a single machine with one thread. And there is a second machine and a second query, running independently of the first machine and the first query.

Is the query an example of:

1. Inter-query parallelism,
2. Intra-query, Inter-operator parallelism
3. Intra-query, Intra-operator parallelism
4. No parallelism

Worksheet

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.

Is the query an example of:

1. Inter-query parallelism,
2. Intra-query, Inter-operator parallelism
3. Intra-query, Intra-operator parallelism
4. No parallelism

Worksheet

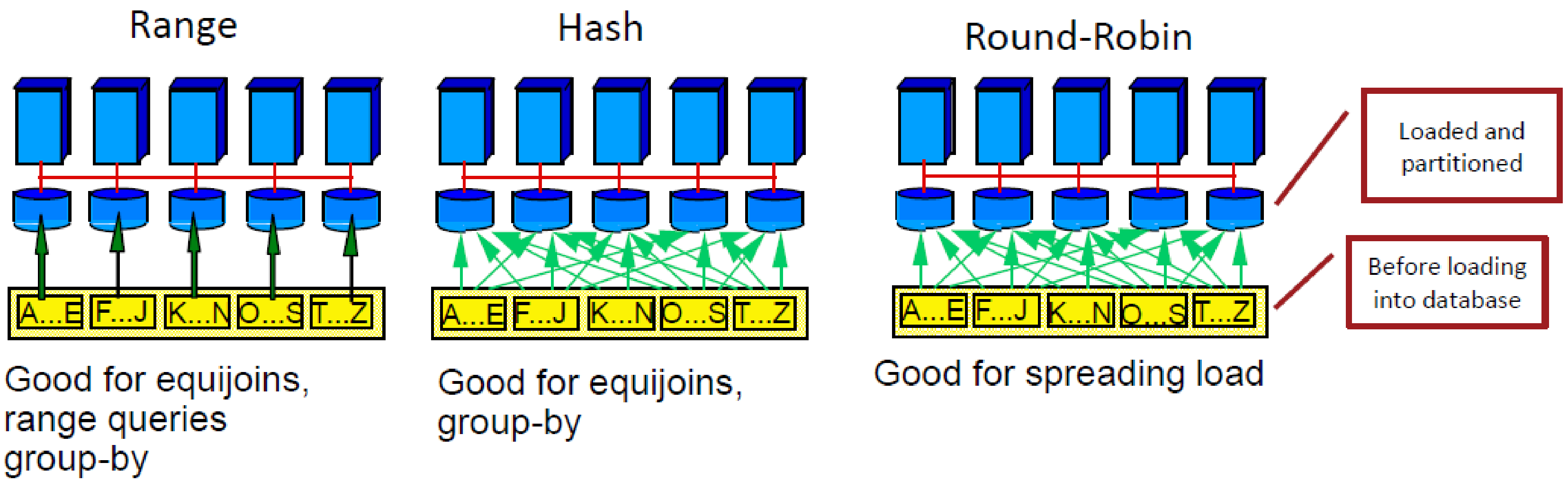
A query with a selection, followed by a projection, runs on a single machine with multiple threads; one thread is given to the selection and one thread is given to the projection.

Is the query an example of:

1. Inter-query parallelism,
2. Intra-query, Inter-operator parallelism
3. Intra-query, Intra-operator parallelism
4. No parallelism

Intra-op Parallelism-Data Partitioning

- **range partitioning**
  - divides data based on which range the key belongs to
- **hash partitioning**
  - divides data based on a hash function
- **round robin partitioning**
  - cycles through the partitions in order as data come in (not ordered based on a key)



Range

Hash

Round-Robin

Loaded and partitioned

Before loading into database

Good for equijoins, range queries group-by

Good for equijoins, group-by

Good for spreading load

WorkSheet

- 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?

WorkSheet

- 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?

Answer:

- Range partitioning also works, because items with the same key still end up on the same machine as required.
- Round-robin partitioning does not do that, so it does not work.

Most DB operations can be done Partition Parallel

1. Partition the data over machines with one of the three methods
2. Perform operation on each machine independently

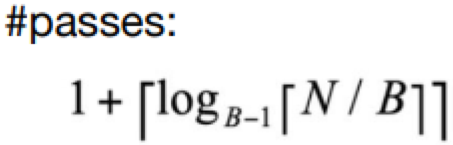
- Parallel Hashing
- Parallel Hash Joins
- Parallel Sorting
- Parallel Sort Merge Join
- ……

Complex plans

- Allow for pipeline parallelism, but sorts, hashes block the pipeline
- Partition parallelism achieved via bushy trees.

WorkSheet

- 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?

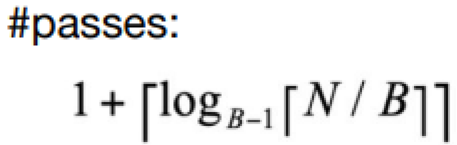


#passes:

$$1 + \lceil \log_{B-1} \lceil N / B \rceil \rceil$$

WorkSheet

- 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?



#passes:

$$1 + \lceil \log_{B-1} \lceil N / B \rceil \rceil$$

- 2 passes
- After range partitioning, each table will have 5,000 KB of data, or 1,250 pages. With 100 pages of buffer, this will take 2 passes to sort.

Extra Things to Note

- We want to calculate the **network cost** in terms of **time**, **or amount of data sent (KB, …)** between machines
  - Less important is the number of I/Os on a given machine.
- Machines may have to receive data from other machines **before** starting processing data if a table is sorted on only a single machine for example.
- Since we have multiple machines to use, we now care about **bottlenecks**.
  - Uneven number of records on each machine causes the total time spent doing operations (scanning, sorting, etc.) to be the **maximum time spent of each individual machine** (Machine 1 takes 500ms and Machine 2 takes 300ms, then our overall parallel query takes 500ms)

WorkSheet

- Suppose we have 4 machines. 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.
- 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.

WorkSheet

- Suppose we have 4 machines. 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.
- 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.

Answer: 25 seconds

- After we uniformly partition our data, Machine 1 will send 25 pages to Machines 2, 3, and 4. It will take 25 seconds to finish sending these pages to each machine if we send the pages to each machine at the same time.

WorkSheet

- Students table consists of 100 pages. Imagine that there is another table, Classes, which is 10 pages. Using just one machine with 10 buffer pages, how long would a BNLJ take if each disk access (read or write) takes 0.5 seconds?
- (BNLJ require [C] + ceil([C] / (B-2))[S] I/Os)

WorkSheet

- Students table consists of 100 pages. Imagine that there is another table, Classes, which is 10 pages. Using just one machine with 10 buffer pages, how long would a BNLJ take if each disk access (read or write) takes 0.5 seconds?
- (BNLJ require [C] + ceil([C] / (B-2))[S] I/Os)

- Answer: 105 seconds
- [C] = 10, [S] = 100, B = 10
- BNLJ will require 10 + ceil(10/8) * 100 = 210 I/Os
- take 210 * 0.5 = 105 seconds.