CSE 344: Section 6
Parallel Databases
Midterm Review

May 3rd, 2018

Agenda for Today

This section:

- Quick touch up on parallel databases
- Midterm review

Distributed Query Processing

In this class, only **shared-nothing architecture** and **intra-operator parallelism**

Horizontal Data Partitioning:

- Block Partition
- Hash partitioned on attribute A
- Range partitioned on attribute A

Moving Data

We have a “network layer” to move tuples temporarily between nodes.

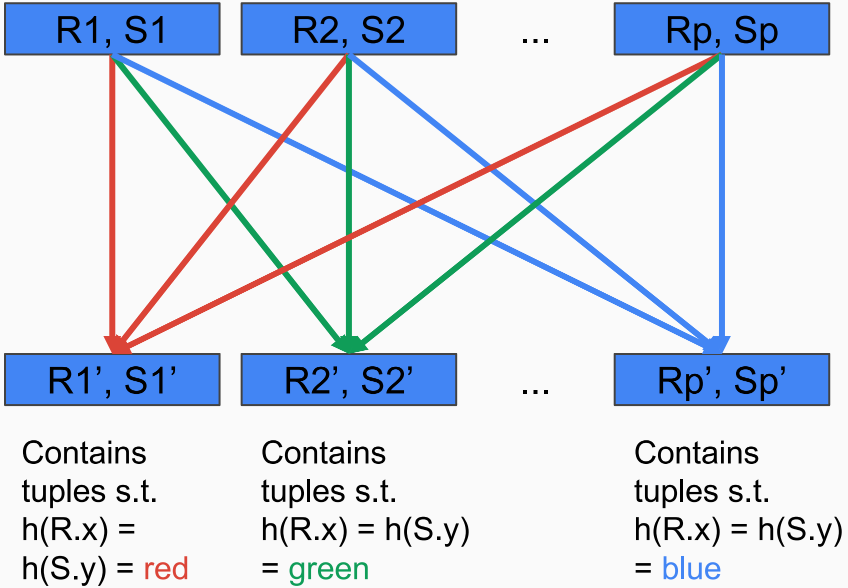
Networking is expensive so we need to be efficient (especially on joins and grouping).

Moving Data:
Partitioned Hash-Join Mechanism

We have p machines

We wish to join on some attribute (say R.x and S.y)

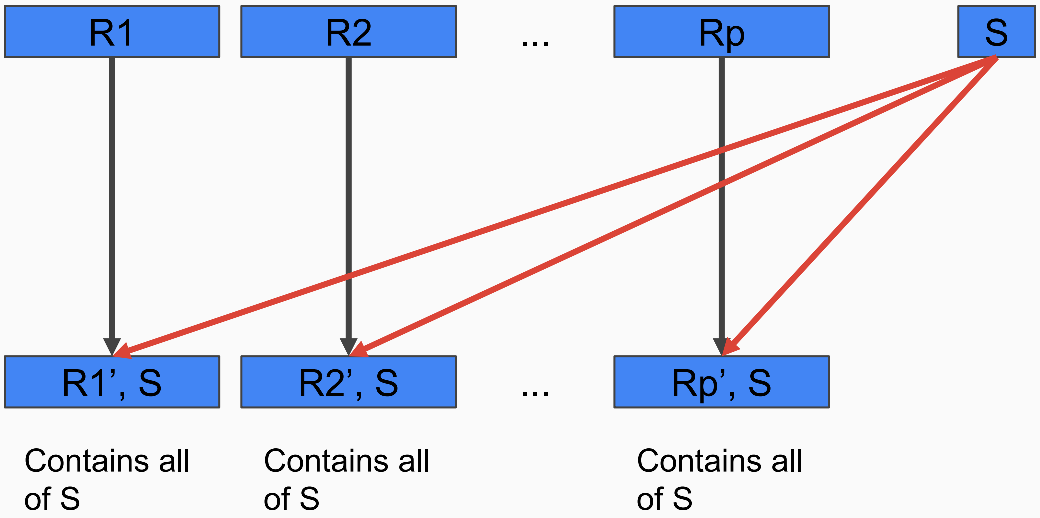
Call our hash function h(z)



Moving Data:
Broadcast Join (Map-Side Join) Mechanism

We want to think about how to prevent sending all data through the network.

Take advantage of small datasets (meaning the whole dataset can fit into main memory)

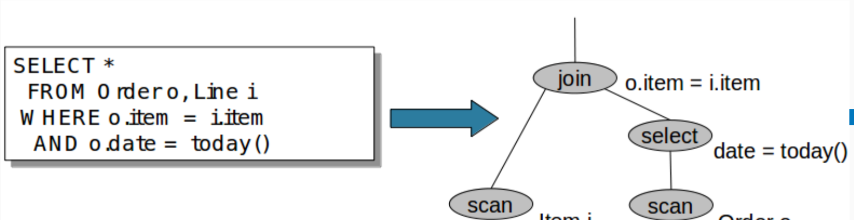


Now What?

“Cool. I know how to split data up and move it around efficiently. What does that have to do with my queries?”

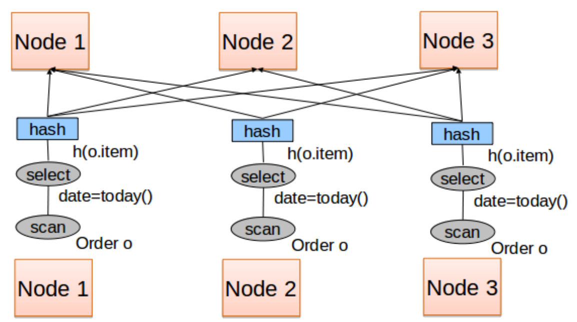
Now What?

"Cool. I know how to split data up and move it around efficiently. What does that have to do with my queries?"



Query

Single Node Plan

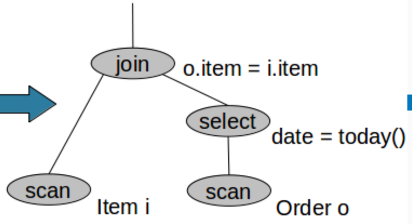


Multi-Node Plan

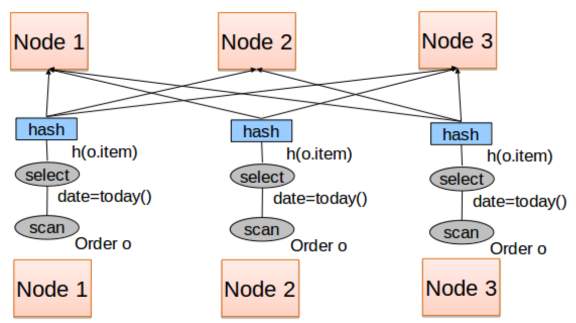
Parallel Query Plans

```
SELECT *
  FROM Order o, Line i
 WHERE o.item = i.item
   AND o.date = today()
```

Query



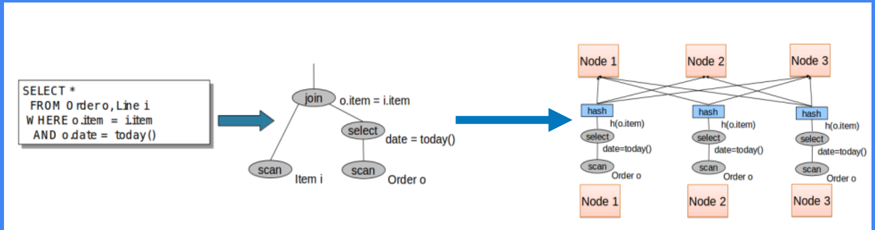
Single Node Plan



Multi-Node Plan

Know how to derive parallel plans though this pipeline.

Parallel Query Plans



Know how to derive parallel plans from your single node plans.

- Which RA operations can you do without talking to other nodes?
- Which RA operations require moving tuples?
- Can we take advantage of how our data is already stored? (partitioning)

$\bowtie$ σ π

Parallel DB Practice!

We have a distributed database that hold the relations:
Drug(spec VARCHAR(255), compatibility INT)
Person(name VARCHAR(100) PK, compatibility INT)

We want to compute: *This is a pretty hard question*

```
SELECT P.name, count(D.spec)
  FROM Person AS P, Drug AS D
 WHERE P.compatibility = D.compatibility
 GROUP BY P.name;
```

Drug is block-partitioned
Person is hash-partitioned on compatibility [h(n)]
You have three nodes. **Draw a parallel query plan.**

$\gamma_{P.name,\ count(D.spec)}(P \ltimes D)$

Take advantage of:

1. Hash partitioning of Drug
2. The PK uniqueness of name