

Introduction to Data Management CSE 344

Lectures 23: Parallel Databases

CSE 344 - Winter 2016

1

Announcement

- WQ7 due tonight (last one!)
- HW7 due on Monday

CSE 344 - Winter 2016

2

Outline

- Finish transactions (last lecture)
- Parallel databases

CSE 344 - Winter 2016

3

Why compute in parallel?

- Multi-cores:
 - Most processors have multiple cores
 - This trend will increase in the future
- Big data: too large to fit in main memory
 - Distributed query processing on 100x-1000x servers
 - Widely available now using cloud services

CSE 344 - Winter 2016

4

Big Data

- Companies, organizations, scientists have data that is **too big, too fast, and too complex** to be managed without changing tools and processes
- Complex data processing:
 - Decision support queries (SQL w/ aggregates)
 - Machine learning (adds linear algebra and iteration)

CSE 344 - Winter 2016

5

Two Kinds to Parallel Data Processing

- **Parallel databases**, developed starting with the 80s (this lecture)
 - **OLTP** (Online Transaction Processing)
 - **OLAP** (Online Analytic Processing, or Decision Support)
- **General purpose distributed processing:** MapReduce, Spark
 - Mostly for **Decision Support Queries**

CSE 344 - Winter 2016

6

Performance Metrics for Parallel DBMSs

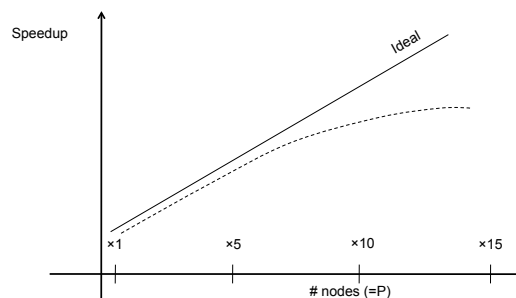
P = the number of nodes (processors, computers)

- **Speedup:**
 - More nodes, same data \rightarrow higher speed
- **Scaleup:**
 - More nodes, more data \rightarrow same speed
- **OLTP:** “Speed” = transactions per second (TPS)
- **Decision Support:** “Speed” = query time

CSE 344 - Winter 2016

7

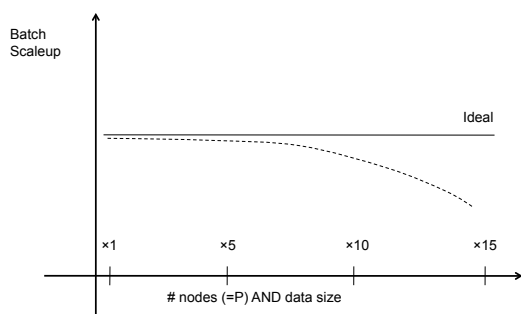
Linear v.s. Non-linear Speedup



CSE 344 - Winter 2016

8

Linear v.s. Non-linear Scaleup



CSE 344 - Winter 2016

9

Challenges to Linear Speedup and Scaleup

- **Startup cost**
 - Cost of starting an operation on many nodes
- **Interference**
 - Contention for resources between nodes
- **Skew**
 - Slowest node becomes the bottleneck

CSE 344 - Winter 2016

10

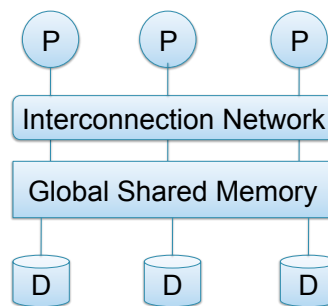
Architectures for Parallel Databases

- Shared memory
- Shared disk
- Shared nothing

CSE 344 - Winter 2016

11

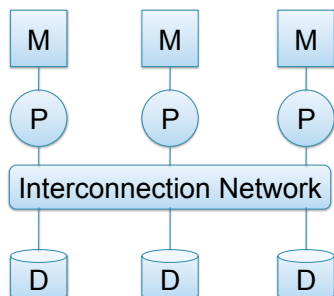
Shared Memory



CSE 344 - Winter 2016

12

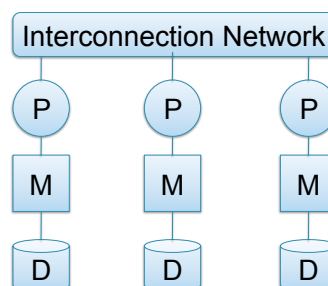
Shared Disk



CSE 344 - Winter 2016

13

Shared Nothing

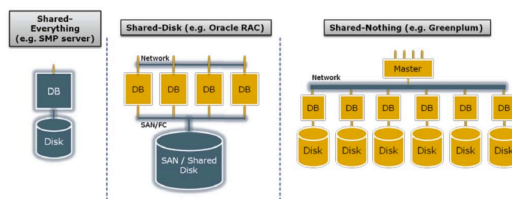


CSE 344 - Winter 2016

14

A Professional Picture...

Figure 1 - Types of database architecture



From: Greenplum (now EMC) Database Whitepaper

SAN = "Storage Area Network"

CSE 344 - Winter 2016

15

Shared Memory

- Nodes share both RAM and disk
- Dozens to hundreds of processors

Example: SQL Server runs on a single machine and can leverage many threads to get a query to run faster (see query plans)

- Easy to use and program
- But very expensive to scale: last remaining cash cows in the hardware industry

CSE 344 - Winter 2016

16

Shared Disk

- All nodes access the same disks
- Found in the largest "single-box" (non-cluster) multiprocessors

Oracle dominates this class of systems.

Characteristics:

- Also hard to scale past a certain point: existing deployments typically have fewer than 10 machines

CSE 344 - Winter 2016

17

Shared Nothing

- Cluster of machines on high-speed network
- Called "clusters" or "blade servers"
- Each machine has its own memory and disk: lowest contention.

NOTE: Because all machines today have many cores and many disks, then shared-nothing systems typically run many "nodes" on a single physical machine.

Characteristics:

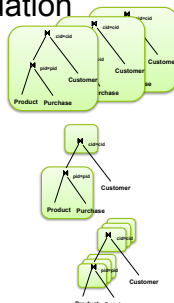
- Today, this is the most scalable architecture.
- Most difficult to administer and tune.

We discuss only Shared Nothing in class

18

Approaches to Parallel Query Evaluation

- **Inter-query parallelism**
 - Transaction per node
 - OLTP
- **Inter-operator parallelism**
 - Operator per node
 - Both OLTP and Decision Support
- **Intra-operator parallelism**
 - Operator on multiple nodes
 - Decision Support



We study only intra-operator parallelism: most scalable

Single Node Query Processing (Review)

Given relations $R(A,B)$ and $S(B,C)$, **no indexes**:

- **Selection:** $\sigma_{A=123}(R)$
 - Scan file R, select records with $A=123$
- **Group-by:** $\gamma_{A, \text{sum}(B)}(R)$
 - Scan file R, insert into a hash table using attr. A as key
 - When a new key is equal to an existing one, add B to the value
- **Join:** $R \bowtie S$
 - Scan file S, insert into a hash table using attr. B as key
 - Scan file R, probe the hash table using attr. B

CSE 344 - Winter 2016

20

Distributed Query Processing

- Data is horizontally partitioned on many servers
- Operators may require data reshuffling

CSE 344 - Winter 2016

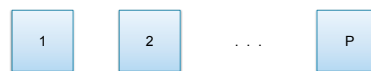
21

Horizontal Data Partitioning

Data:

K	A	B
...
...
...
...
...
...
...
...
...

Servers:



CSE 344 - Winter 2016

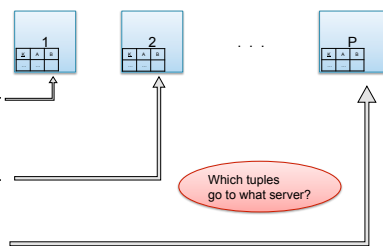
22

Horizontal Data Partitioning

Data:

K	A	B
...
...
...
...
...
...
...
...
...

Servers:



CSE 344 - Winter 2016

23

Horizontal Data Partitioning

- **Block Partition:**
 - Partition tuples arbitrarily s.t. $\text{size}(R_1) \approx \dots \approx \text{size}(R_P)$
- **Hash partitioned on attribute A:**
 - Tuple t goes to chunk i , where $i = h(t.A) \bmod P + 1$
- **Range partitioned on attribute A:**
 - Partition the range of A into $-\infty = v_0 < v_1 < \dots < v_P = \infty$
 - Tuple t goes to chunk i , if $v_{i-1} < t.A < v_i$

CSE 344 - Winter 2016

24

Parallel GroupBy

Data: $R(\underline{K}, A, B, C)$

Query: $\gamma_{A, \text{sum}(C)}(R)$

Discuss in class how to compute in each case:

- R is hash-partitioned on A
- R is block-partitioned
- R is hash-partitioned on K

CSE 344 - Winter 2016

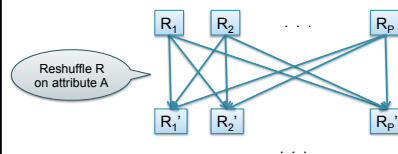
25

Parallel GroupBy

Data: $R(\underline{K}, A, B, C)$

Query: $\gamma_{A, \text{sum}(C)}(R)$

- R is block-partitioned or hash-partitioned on K



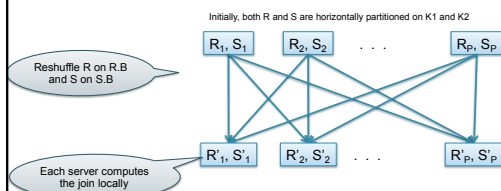
CSE 344 - Winter 2016

26

Parallel Join

• **Data:** $R(\underline{K1}, A, B), S(\underline{K2}, B, C)$

• **Query:** $R(\underline{K1}, A, B) \bowtie S(\underline{K2}, B, C)$



CSE 344 - Winter 2016

27

Data: $R(\underline{K1}, A, B), S(\underline{K2}, B, C)$

Query: $R(\underline{K1}, A, B) \bowtie S(\underline{K2}, B, C)$

Partition

R1		S1		R2		S2	
K1	B	K2	B	K1	B	K2	B
1	20	101	50	3	20	201	20
2	50	102	50	4	20	202	50

Shuffle

R1'		S1'		R2'		S2'	
K1	B	K2	B	K1	B	K2	B
1	20	201	20	2	50	101	50
3	20					102	50
4	20					202	50

Local Join

CSE 344 - Winter 2016

28

Speedup and Scaleup

- Consider:
 - Query: $\gamma_{A, \text{sum}(C)}(R)$
 - Runtime: dominated by reading chunks from disk
- If we double the number of nodes P , what is the new running time?
 - Half (each server holds $\frac{1}{2}$ as many chunks)
- If we double both P and the size of R , what is the new running time?
 - Same (each server holds the same # of chunks)

CSE 344 - Winter 2016

29

Uniform Data v.s. Skewed Data

- Let $R(\underline{K}, A, B, C)$; which of the following partition methods may result in **skewed** partitions?

- Block partition

Uniform

- Hash-partition

– On the key K

Uniform

– On the attribute A

May be skewed

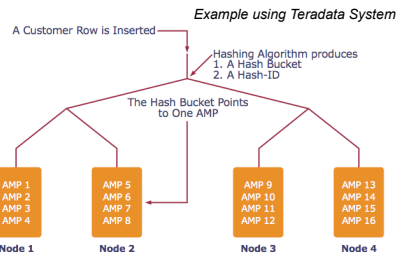
Assuming good hash function

E.g. when all records have the same value of the attribute A , then all records end up in the same partition

CSE 344 - Winter 2016

30

Loading Data into a Parallel DBMS



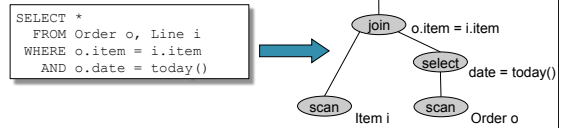
AMP = "Access Module Processor" = unit of parallelism

CSE 344 - Winter 2016

31

Example Parallel Query Execution

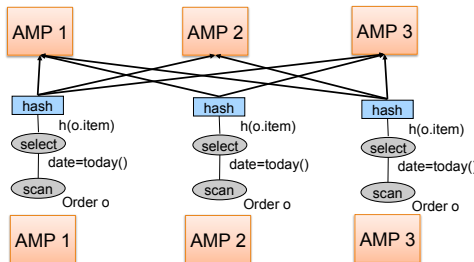
Find all orders from today, along with the items ordered



CSE 344 - Winter 2016

32

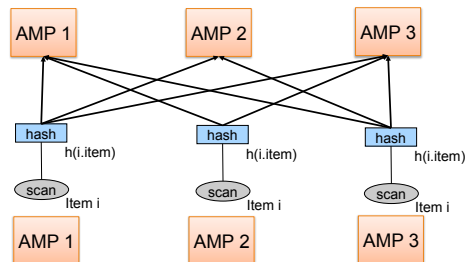
Example Parallel Query Execution



CSE 344 - Winter 2016

33

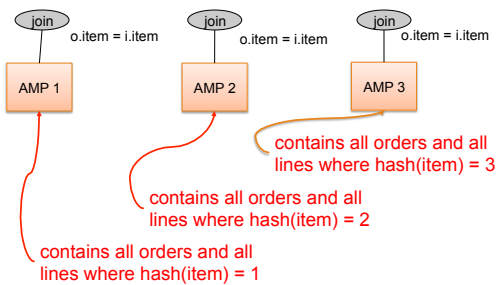
Example Parallel Query Execution



CSE 344 - Winter 2016

34

Example Parallel Query Execution



CSE 344 - Winter 2016

35