Parallel Databases

R&G Chapter 22

(slides adapted from content by J.Gehrke, J.Shanmugasundaram, and/or C.Koch)

Why Scale Up?

Scan of 1 PB at 300MB/s (SATA r2 Limit)

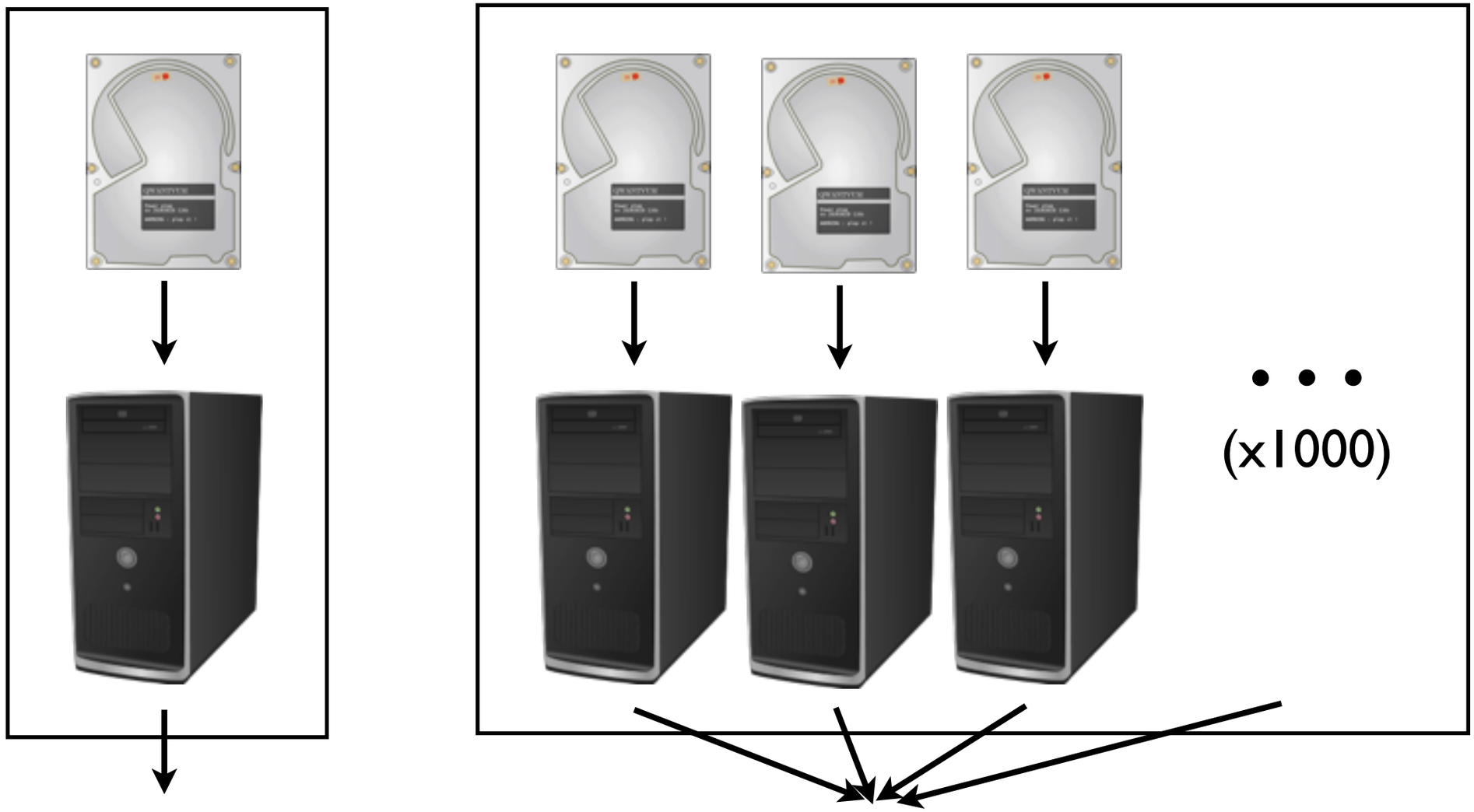


image credit: openclipart.org

Examples of Scaling Up

- Distributed Databases
  - Teradata, Greenplum, HStore, Dremel
- Big Data Processing
  - Map/Reduce (Pig Latin, etc…), Cassandra
- Transaction Processing
  - PNuts, Spanner, Percolator

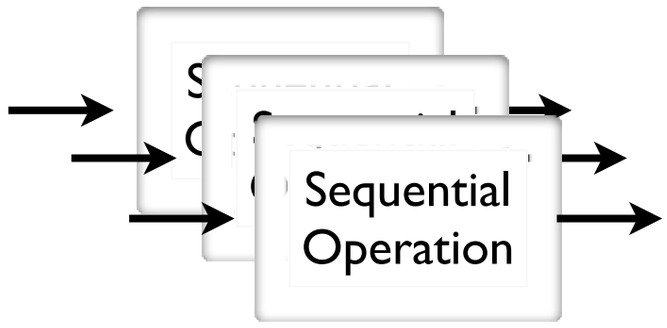
The "Death" of Moore's Law

- Processor speeds have stopped growing.
- But # of transistors still growing!
  - More cores, not faster cores.
  - Need ||-ism to exploit additional cores!
- Lower-power devices are slower, but we can have many more of them!

Types of Parallelism

- Pipeline Parallelism: A task breaks down into stages; each machine processes one stage.

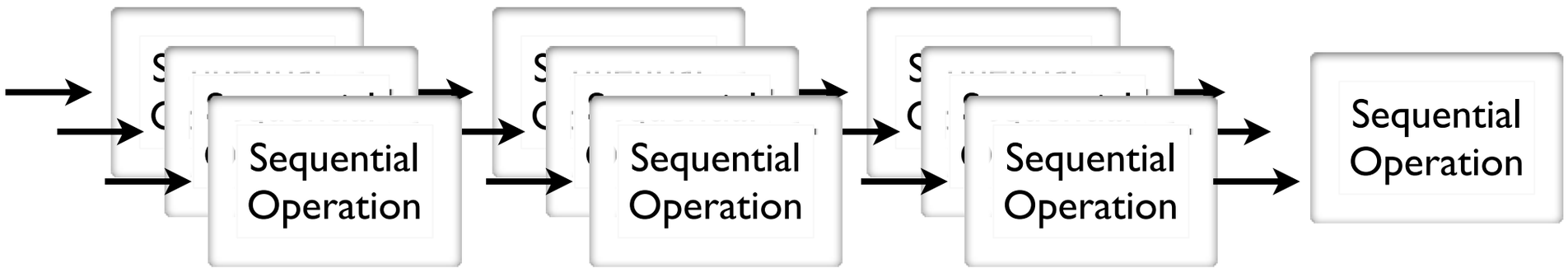
- Partition Parallelism: Many machines doing the same thing to different pieces of data.



Types of Parallelism

- Both types of parallelism are natural in a database management system.

```
SELECT SUM(…) FROM Table WHERE …
```



LOAD SELECT AGG Combine

DBMSes: The First || Success Story

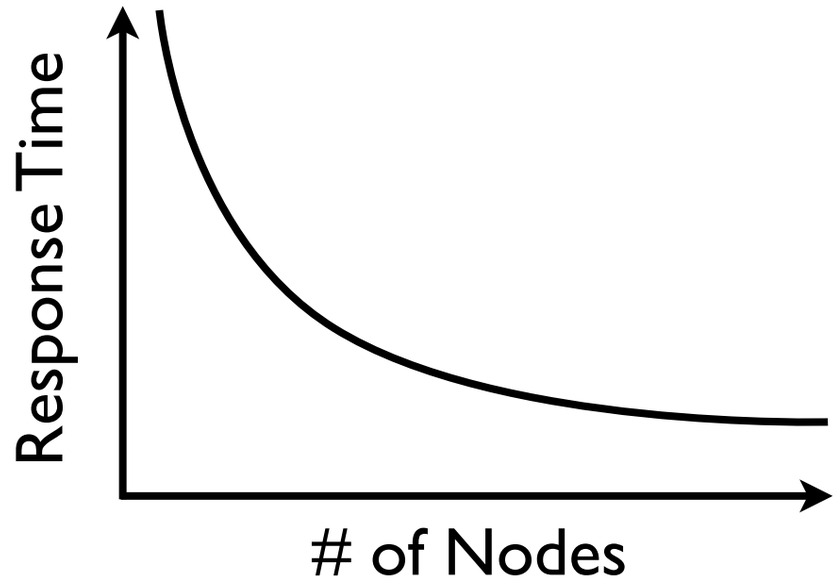
- Every major DBMS vendor has a || version.
- Reasons for success:
  - Bulk Processing (Partition ||-ism).
  - Natural Pipelining in RA plan.
  - Users don't need to think in ||.

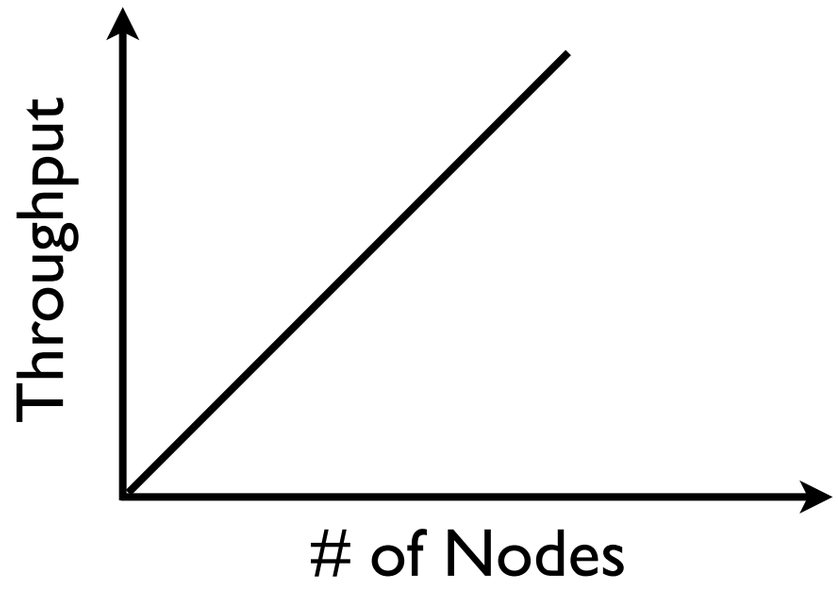
DBMSes: The First || Success Story

- All-in-one DBMSes fail at >100s of nodes.
- Reasons for failure:
  - Too many features (Regress to NoSQL).
  - Too strong consistency.
  - Legacy optimizers not tuned for >1000s of nodes and network limitations.
  - Data curation doesn't scale.

|| Terminology

- Speed-up ||-ism
  - More resources = proportionally less time spent.
- Scale-up ||-ism
  - More resources = proportionally more data processed.





|| Architectures
Shared Memory (SMP)

Clients

CPU

Memory

Disk

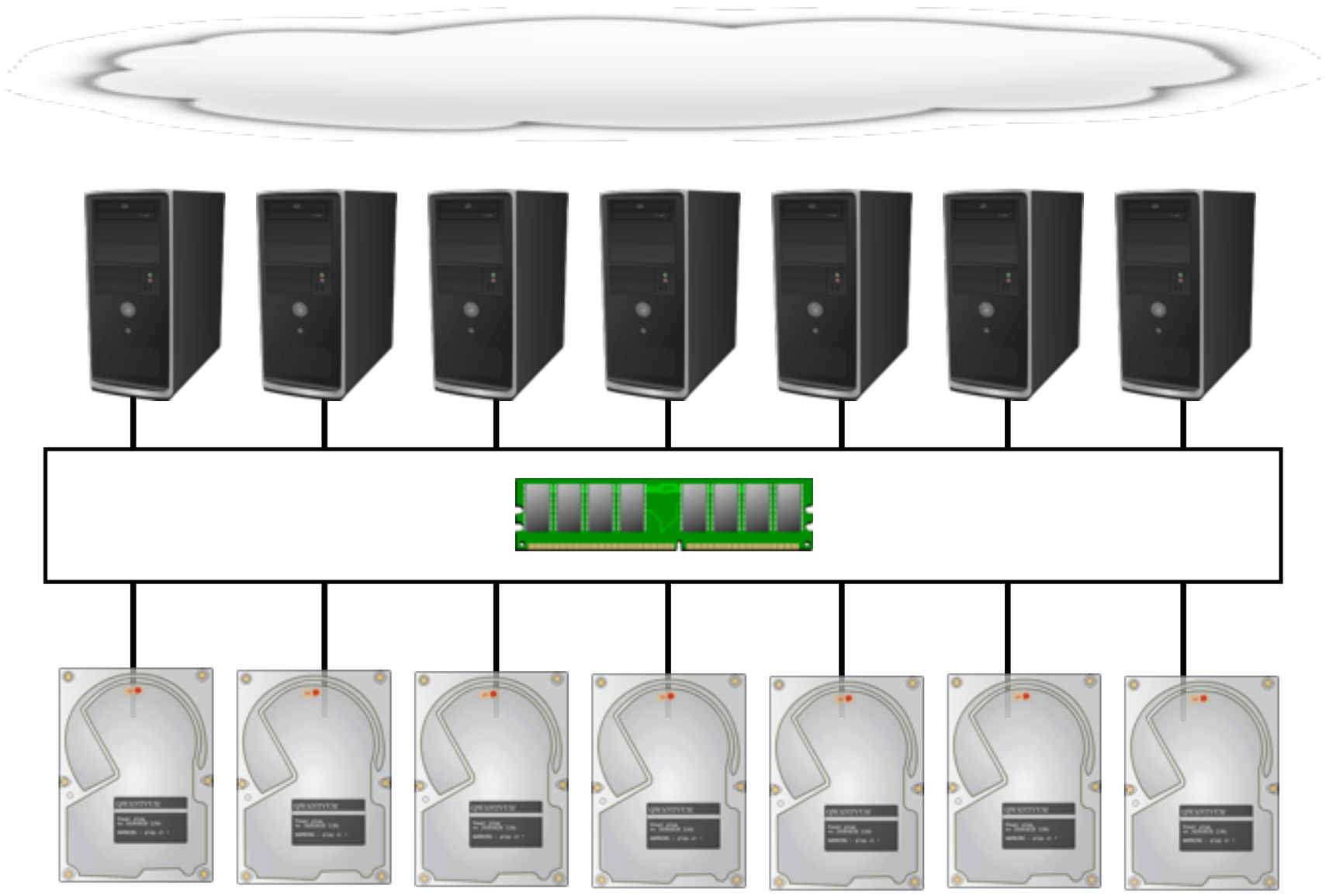


image credit: openclipart.org

|| Architectures
Shared Memory (SMP)

- Examples: Sun, SGI, Sequent
- Advantages
  - Easy to Program
- Disadvantages
  - Expensive to Build
  - Hard to Scale (Not used for ‘Big Data’)

Academic Systems: 100s of Nodes; Deployed: 10s of nodes

|| Architectures
Shared Disk/Storage

Clients

CPU + Memory

Disk

image credit: openclipart.org

|| Architectures
Shared Disk/Storage

- Examples: HDFS/GFS, PNuts, EC2 SimpleDB
- Advantages
  - Can Program Without Considering Parallelism
- Disadvantages
  - Hard to Provide ACID Consistency Efficiently.

Strong Consistency: 100s of Nodes
Weak Consistency: 1000s+ of Nodes

|| Architectures
Shared Nothing

Clients

CPU +
Memory

Disk

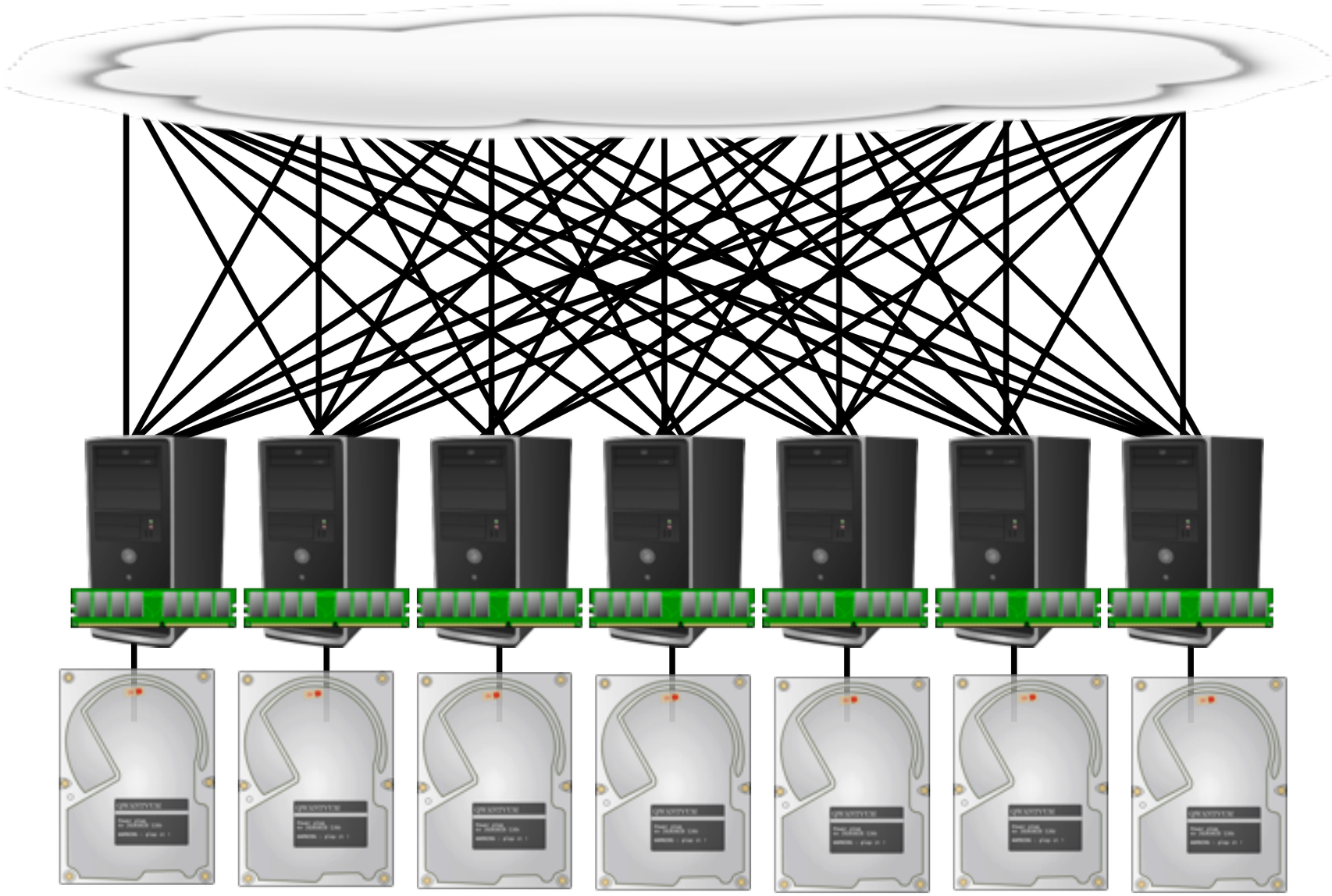


image credit: openclipart.org

|| Architectures
Shared Nothing

- Examples: 'Map/Reduce', Greenplum, Teradata
- Advantages
  - Cheap to Build
  - Easy to Scale
- Disadvantages
  - Hard to Program

Strong Consistency: 1000s+ of Nodes

||-ism in the DBMS

- Intra-operator parallelism
  - All machines cooperate to compute an operation (e.g., scan, sort, …)
- Inter-operator parallelism
  - Each operator runs in parallel on a different machine
- Inter-query parallelism
- Our Focus: Intra-Operator Parallelism

Partitioning the Data

- Which machine does a data value end up at?
- Range Partitioning:
  - Split data up into N ranges.
- Hash Partitioning
  - Split data up based on hash value.
- Round Robin
  - Value 1 to machine 1, 2 to machine 2, etc…

Parallel Scans

- Scan the data separately on each machine
- Merge the results together
  - E.g., compute SUM(X) on each machine
  - SUM all computed SUM(X)
- May not need all machines to participate.
- We can build an index at each site.

Parallel Sorts

- Record as of March 2013: TritonSort (UCSD)
  - 0.725 TB/Min using 52 nodes
    - 1 Trillion Records in ~2.5 min
- http://sortbenchmark.org
- http://sortbenchmark.org/2011_06_tritonsort.pdf

Parallel Sorts

- General Strategy
  - Scan in parallel
  - Range partition to other nodes
  - As tuples come in, sort locally
- Problem: Skew!
  - How do we pick partition boundaries?

Parallel Aggregates

- Aggregate Functions:
  - Distributive: F(A,B,C,D) = F(F(A,B),F(C,D))
  - Algebraic: F(A,B,C,D) = G(H(A,B),H(C,D))
  - Holistic: Can't decompose F.
- What are some examples of each?
- How do we exploit this for ||ism?

Parallel Group By

- Aggregate each group in parallel.
- Use Hash fn to push decomposed aggregate values to the same node.

To Be Continued
(Joins, Parallel Execution Plans)