Why compute in parallel?

- Multi-cores:
  - Most processors have multiple cores
  - This trend will likely 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

Parallel DBMSs

- How to evaluate a parallel DBMS?
- How to architect a parallel DBMS?
- How to partition data in a parallel DBMS?

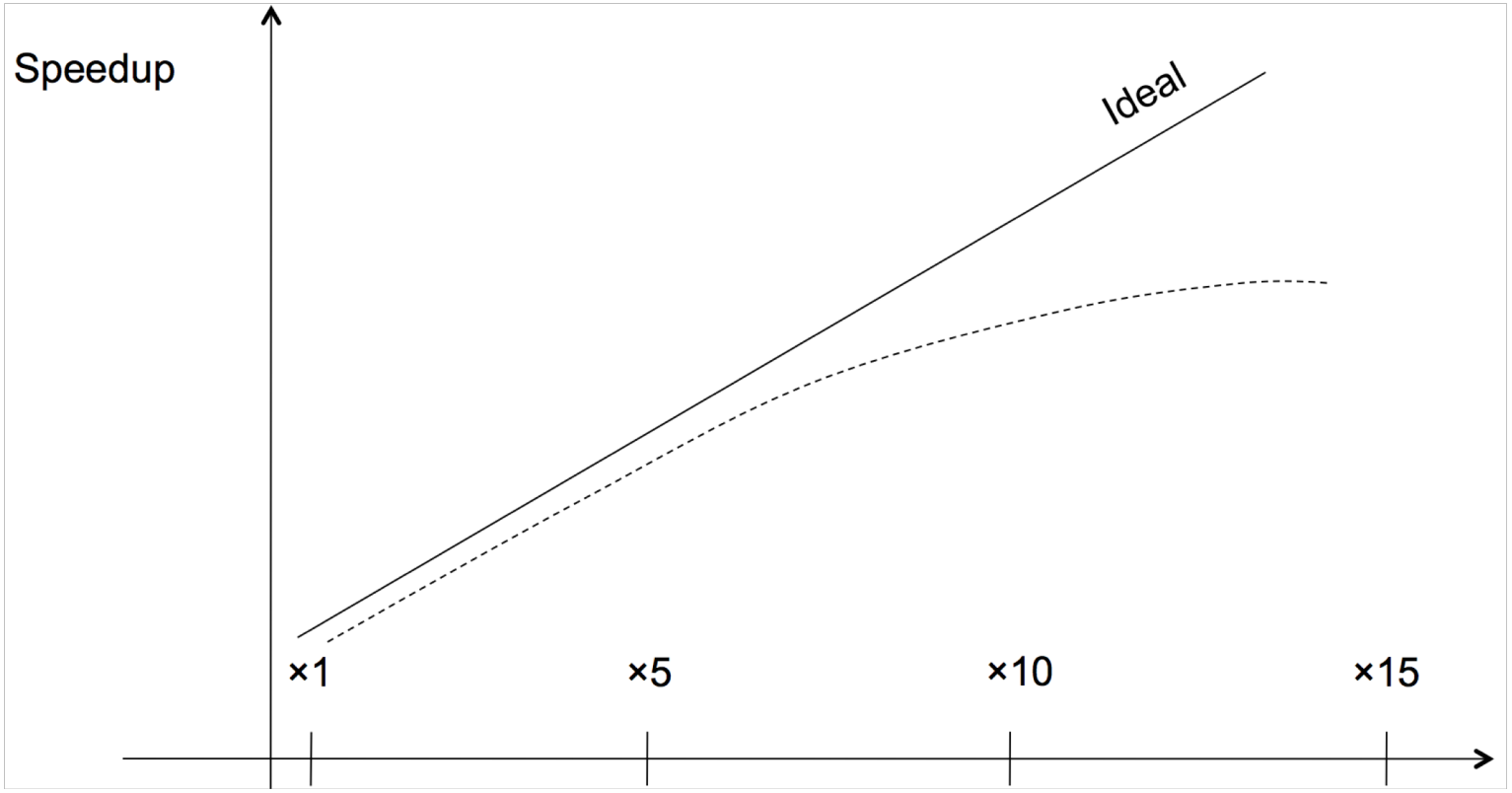
Parallel DBMSs

- **How to evaluate a parallel DBMS?**
- How to architect a parallel DBMS?
- How to partition data in a parallel DBMS?

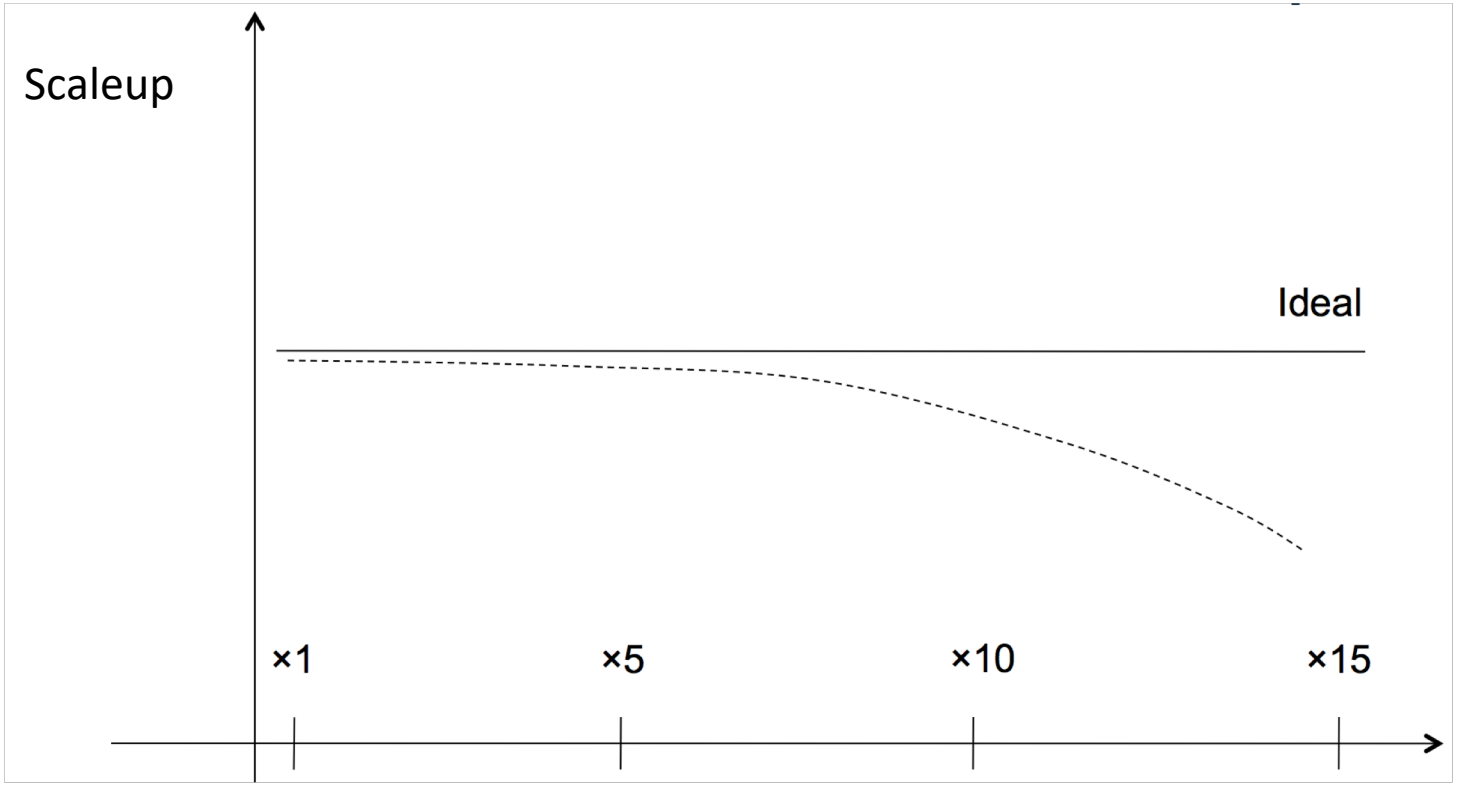
Performance Metrics for Parallel DBMSs

- Nodes = processors, compute
- **Speedup**
  - More nodes, same data → Higher speed
- **Scaleup**
  - More nodes, more data → same speed

Linear v.s. Non-linear Speedup



Linear v.s. Non-linear Scaleup



nodes AND data size

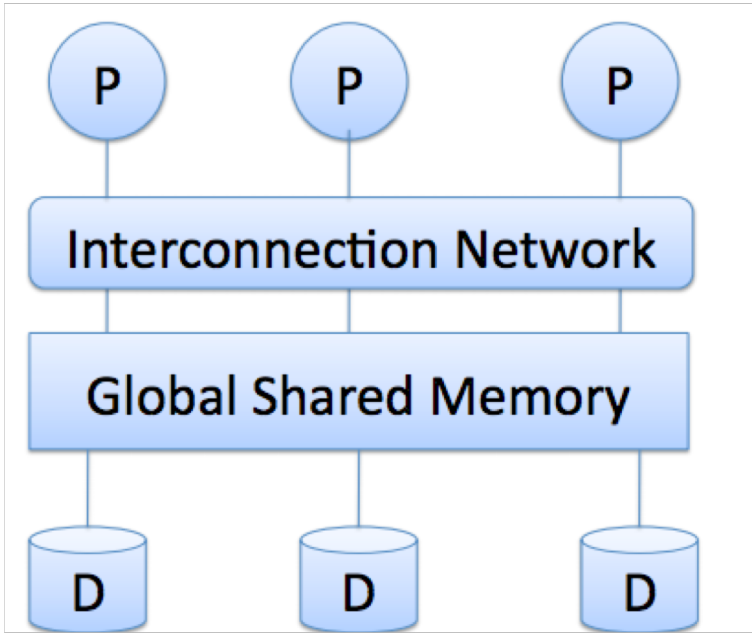
Parallel DBMSs

- **How to evaluate a parallel DBMS?**
- **How to architect a parallel DBMS?**
- How to partition data in a parallel DBMS?

Three Architectures

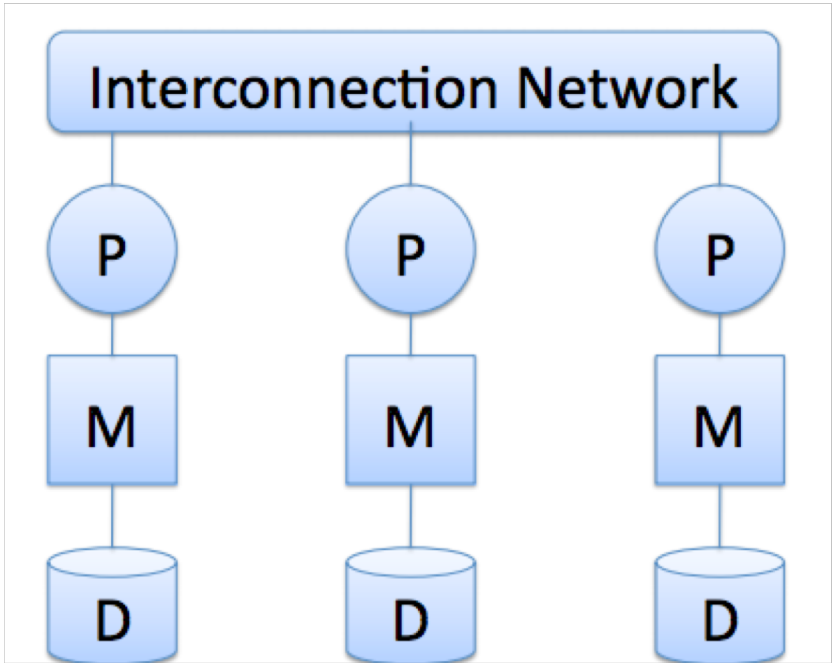
- **Shared Memory**
- **Shared Nothing**
- **Shared Disk**

Shared Memory



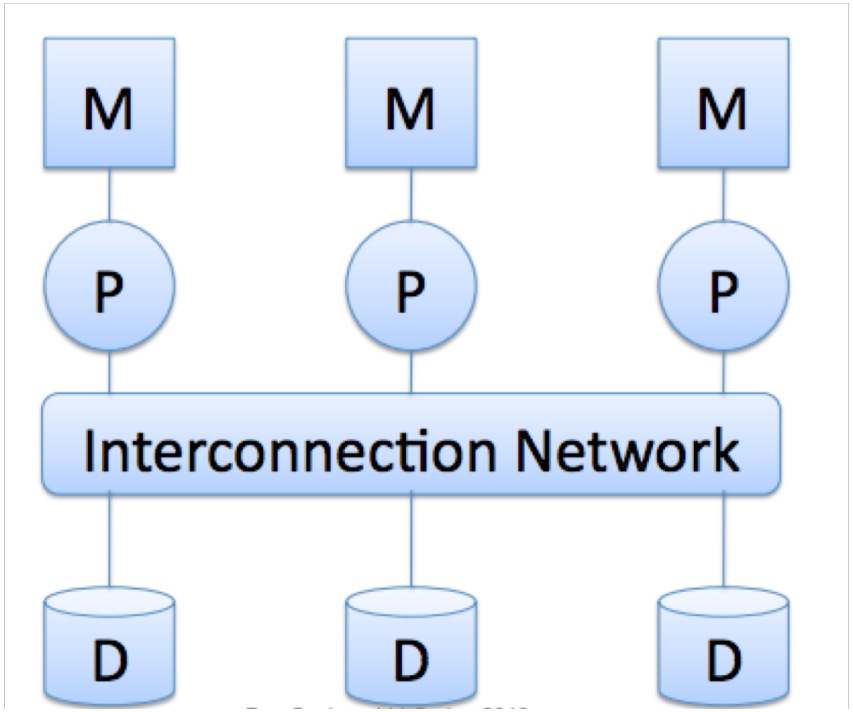
GPU

Shared Nothing



Parallel DBMSs, MapReduce, Spark

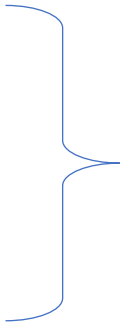
Shared Disk



Azure Data Warehouse

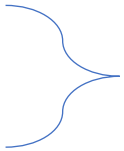
Three Architectures

- **Shared Memory**
- **Shared Nothing**



Computation vs. Communication Trade-offs

- **Shared Disk**



Economic Consideration

Parallel DBMSs

- How to evaluate a parallel DBMS?
- How to architect a parallel DBMS?
- **How to partition data in a parallel DBMS?**

Horizontal Data Partitioning

- **Round Robin**
  - ☺ Load Balancing
  - ☹ Bad Query Performance
- **Range Partitioning**
  - ☺ Good for range/point queries
  - ☹ Data Skew (i.e., Bad Load balancing)
- **Hash Partitioning**
  - ☺ Load Balancing, Good for point queries
  - ☹ Hard to answer range queries