Hadoop/MapReduce: Overall Computing Paradigm

Part 1

Large-Scale Data Analytics

- MapReduce computing paradigm (E.g., Hadoop) vs. Traditional database systems

vs.

- **Enterprises are turning to Hadoop**
  - Especially applications generating ***big data***
  - Web applications, social networks, scientific applications

Why Hadoop is able to compete?

hadoop

vs.

Database

- Scalability (petabytes of data, thousands of machines)
- Flexibility in accepting all data formats (no schema)
- Efficient and simple fault-tolerant mechanism
- Commodity inexpensive hardware

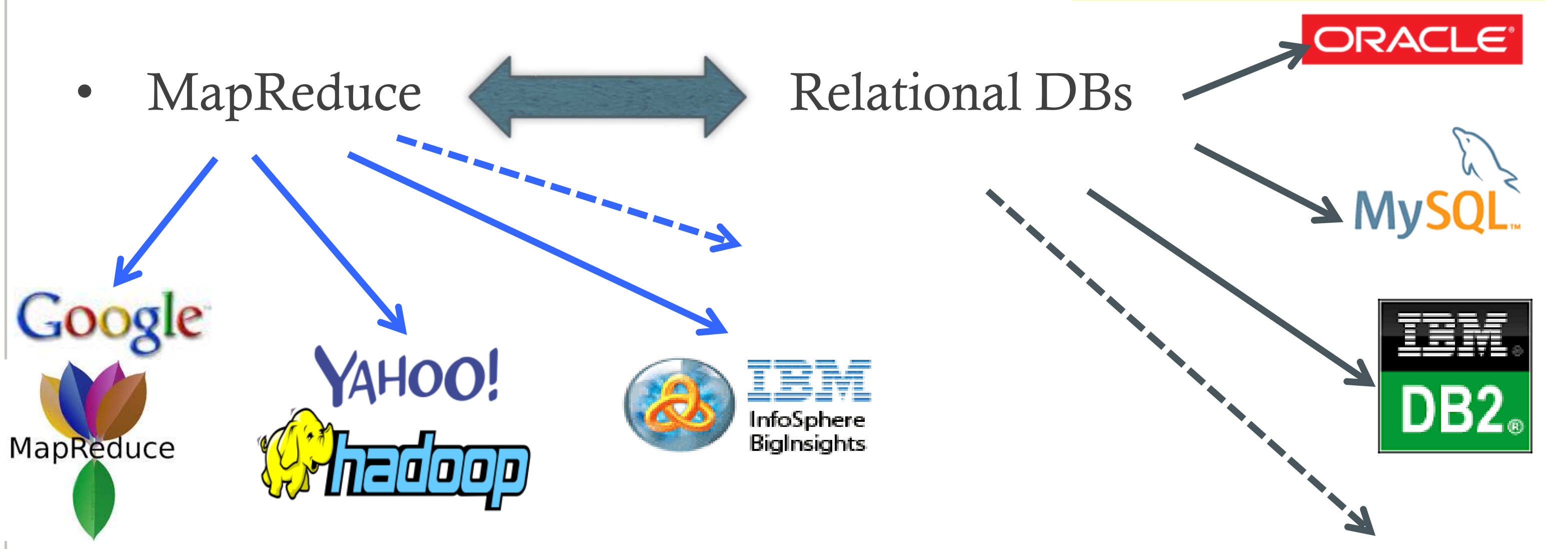
- Performance (tons of indexing, tuning, data organization tech.)
- Fast Processing — Interactive processing
- Transactions and consistency guarantees (ACID)
- Features:
  - Provenance tracking
  - Annotation management
  - ….

What is MapReduce

- **MapReduce is a "computational paradigm"**
  - A specific mechanism of processing the data

Different Implementations/Systems

- MapReduce ⟷ Relational DBs



What is Hadoop ?

- Hadoop is a software framework for *distributed processing* of *large datasets* across *large clusters* of computers
  - ***Large datasets*** → Terabytes or petabytes of data
  - ***Large clusters*** → hundreds or thousands of nodes
- Hadoop is open-source implementation for Google ***MapReduce***
- Hadoop based on simple programming model called *MapReduce*
- Hadoop is based on a simple data model, *any data will fit*

Compute Cluster

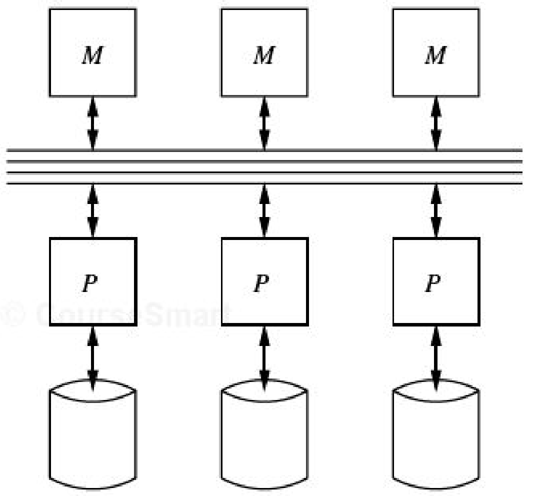
A rack of *N* machines

One machine ←→ One node

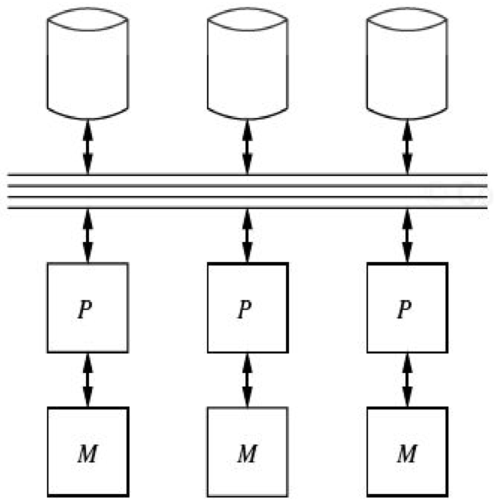
Compute Cluster

- Cluster ➔ Set of machines connected together

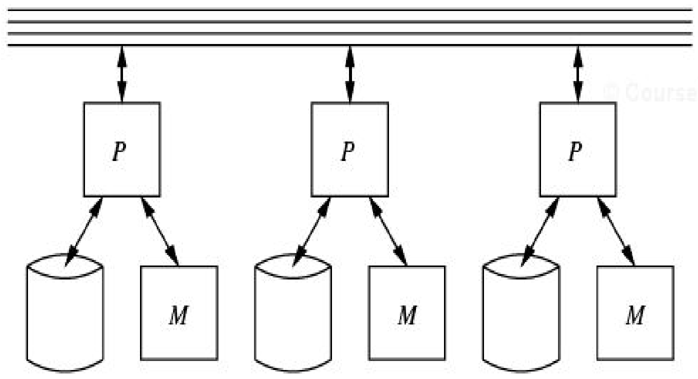
Shared-memory



Shared-disk

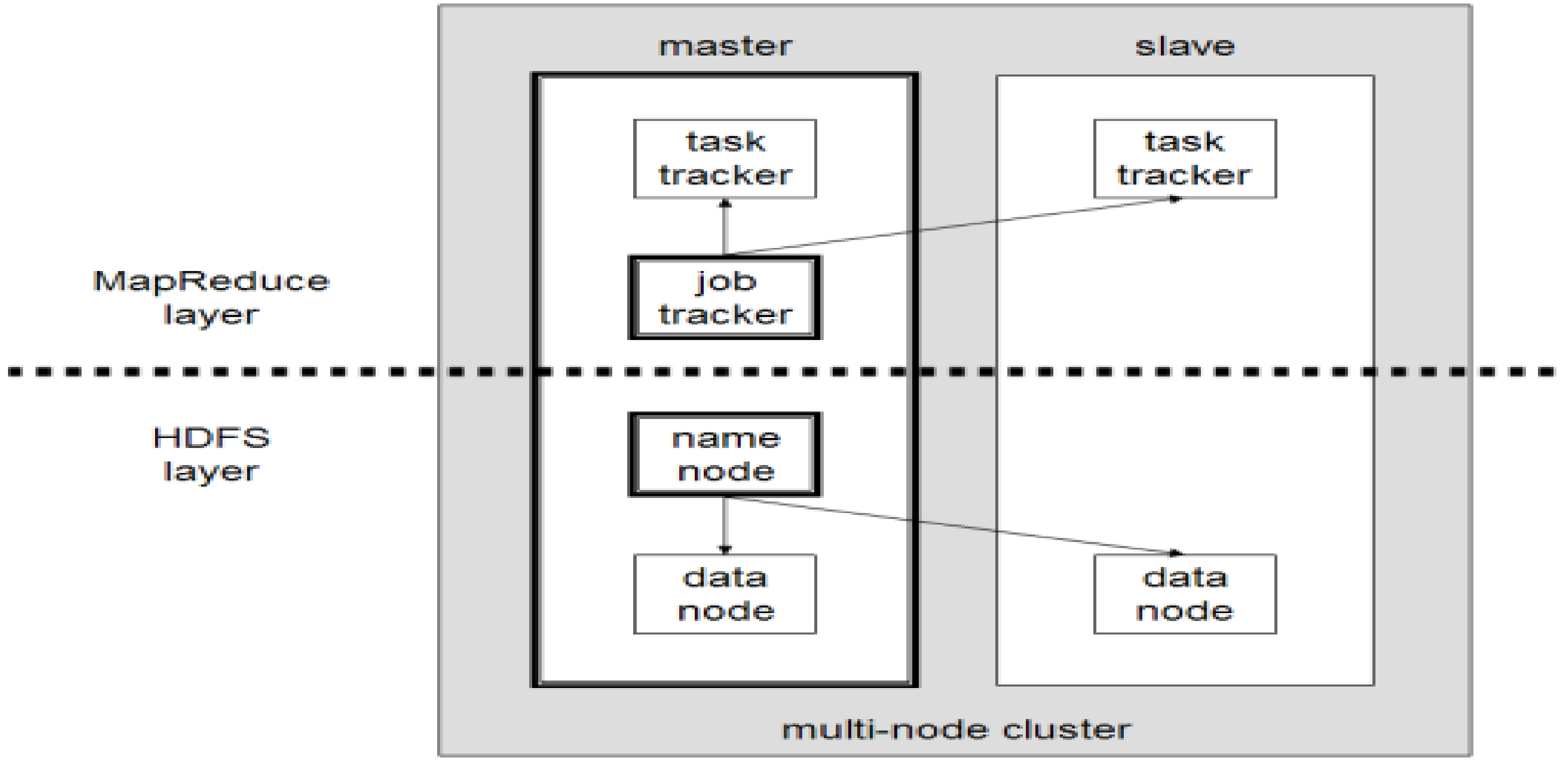


Shared-nothing

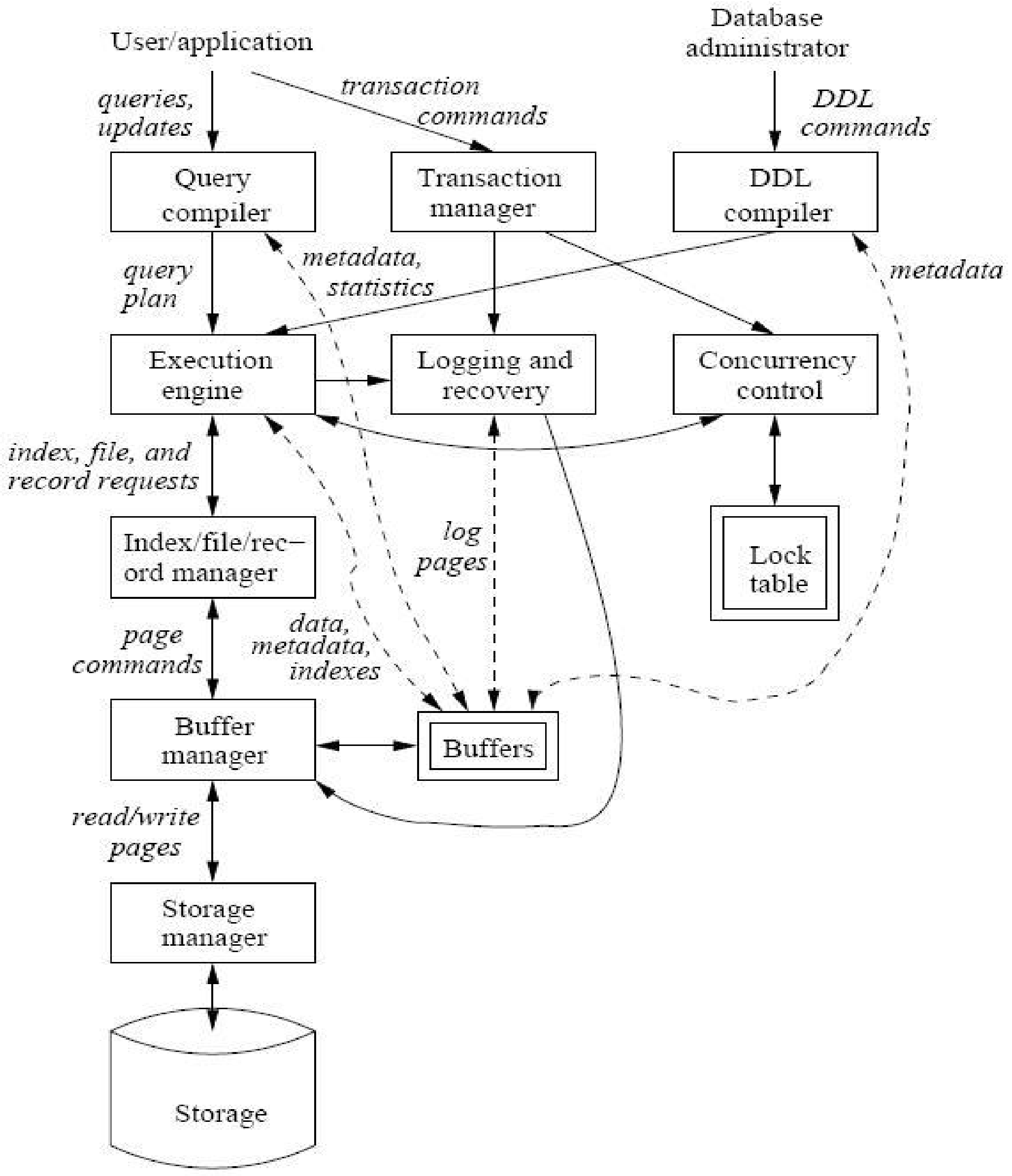


What is Hadoop (Cont'd)

- **Hadoop framework consists of two main layers:**
  - Distributed file system (HDFS)
  - Execution engine (MapReduce)



Contrast it with RDBMS



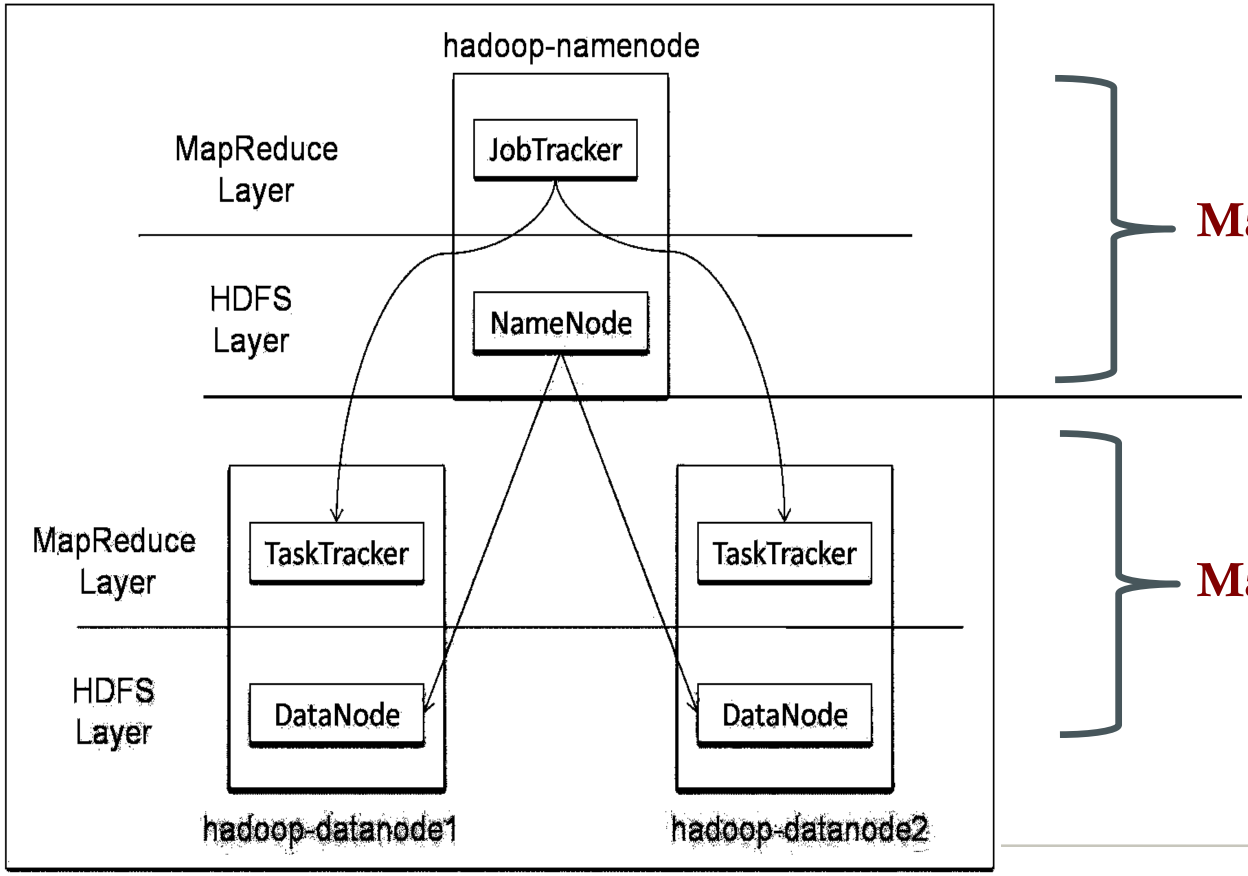
Database management system components

Design Principles of Hadoop

- **Scale to big data**
- **Simple to use**
- **Minimal functionality**

Hadoop Master/Slave Architecture

- Hadoop is designed as *master-slave* *shared-nothing* architecture



Master node (single node)

Many slave nodes

Design Choices of Hadoop

- Need to process big data
- Need to parallelize computation across thousands of nodes
- **Commodity hardware**
  - **Large number of low-end cheap machines working in parallel to solve a computing problem**
- This is in contrast to **Parallel DBs**
  - **Small number of high-end expensive machines**

Design Principles of Hadoop

- **Automatic parallelization & distribution**
  - Hidden from the end-user
- **Fault tolerance and automatic recovery**
  - Nodes/tasks will fail and will recover automatically
- **Clean and simple programming abstraction**
  - Users only provide two functions "map" and "reduce"

Who Uses MapReduce/Hadoop?

- Google: Inventors of MapReduce computing paradigm
- Yahoo: Developing Hadoop open-source of MapReduce
- IBM, Microsoft, Oracle
- Facebook, Amazon, AOL, NetFlix
- Many others ++
- Universities and research labs