



## After this video you will be able to

- Explain the various advantages of using a DBMS over a file system
- Specify the differences between a parallel and distributed file system
- Briefly describe a MapReduce-style DBMS

## Storing Data – Files vs. DBMS

- In the old times, database operations were applications in file systems
- Problems
  - Data redundancy, inconsistency and isolation
  - Each task a program
  - Data integrity
  - Atomicity of updates

# Advantages of a DBMS

- Declarative query languages
  - No more task-based programs
- Data independence
  - Applications don't worry about data storage formats and locations
- Efficient access through optimization
  - The system automatically finds an efficient way to access data

## Advantages of a DBMS

- Data integrity and security
  - Methods to keep the accuracy and consistency of data despite failure
    - ACID properties of transactions
    - Failure recovery

#### Concurrent access

 Many users can simultaneously access data without conflict

## Parallel and Distributed DBMS

### Parallel database system

- Improve performance through parallel implementation
- Often allows data replication
  - Data redundancy against table corruption
  - More concurrent queries

### Distributed database system: !

 Data is stored across several sites, each site managed by a DBMS capable of running independently!

Does your big data problem need these facilities?

# DBMS and MapReduce-style Systems

- Started with a different problem focus
  - DBMSs: efficient storage, transactions and retrieval
    - Partitioned data parallelism
    - Account for computation and communication cost
    - Not node failure
  - Mapreduce-style systems: complex data processing over a cluster of machines
    - HDFS-based
    - Analytics data mining, clustering, machine learning
    - Multi-stage, problem-specific algorithms
    - Operate on wider variety of data including text

## **Shifting Requirements**

- Data loading a new bottleneck
  - Does the application need data sooner than the loading time?
- Too much functionality
  - Does the application use only a few data management features?
- Combined Transactional and Analytical Capabilities

## No Single Solution

#### Mixed solutions

- DBMS on HDFS
  - Hadoop-DBMS interoperation
- Relational operations in MapReduce systems like Spark
- Streaming input to DBMS
- New parallel programming models for analytical computation within DBMS