Databases for Big Data

Common Tasks of Databases

- Store data
- Retrieve small bits of information
  - Quickly fetch individual pieces of data
- Process transactions
  - Changes to database which often need to happen in one group
- Produce Reports and Analysis

How Relational Databases Do

- Store Data
  - For small to moderate amounts of data
    - Great!
  - For really large data sets
    - Too big to fit on one machine
    - Problem!

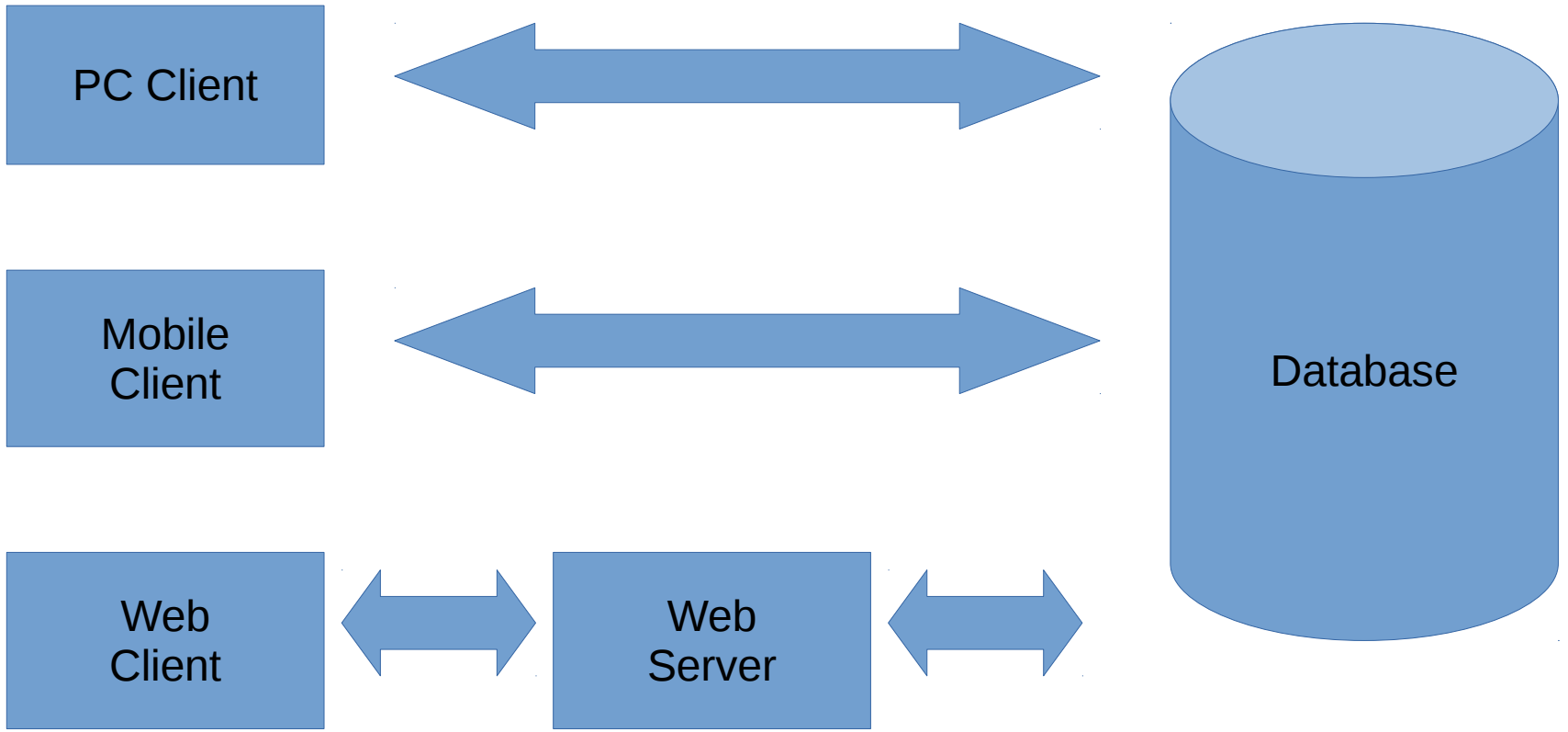
How Relational Databases Do

- Retrieve small bits of information
  - Queries have to be parsed, optimized, and then run
  - Overhead can be a problem
  - Especially when total data is too big for one machine
- Process Transactions
  - Really well
  - But incur overhead to support transactions

How Relational Databases Do

- Produce Reports
  - Pretty well
- Produce Analysis
  - Long-lived queries which have to do serious number crunching are a problem

Database Communication Model



Problems with Single Server

- Data Size
- Reliability
  - What happens if server crashes?
  - What happens if the server’s network is down?
- Computationally Intense Analysis
  - Slows everything down
  - Can’t be parallelized

NoSQL

- Means databases which aren’t oriented around SQL query support
- “Not Only SQL” backronym sometimes used
  - Some NoSQL databases have some SQL support
- Goals:
  - Handle simple data retrieval quicker
  - Distribute data across multiple servers
  - Distribute processing across multiple servers

NoSQL

- Non-relational databases
  - More flexible data representation
  - Fewer data consistency guarantees
- Support query through non-SQL means
  - Usually language-specific libraries
- Limited transaction support
  - Eventually correct rather than always consistent

NoSQL

- Data can be split across servers
  - Allows for much larger sets of data
    - Petabytes or Exabytes
- Redundant server support
  - Better reliability when failures occur

NoSQL Data Models

- Column
- Document
  - XML, JSON
- Key-Value
- Graph
- Object
- Tuple store

How NoSQL Databases Do

- Store Data
  - Great!
- Retrieve small bits of information
  - Great!
- Process Transactions
  - Not good
  - Hard to group changes into transactions
  - Limited guarantees about consistency

How NoSQL Databases Do

- Produce Reports
  - Not so good
  - Reports often require a lot of data correlation
  - Correlating distributed data is slow
- Produce Analysis
  - Depends on particular NoSQL database
    - Some aren’t designed for analysis
  - If using things like Map-Reduce, great

Map-Reduce

- Google algorithm for data analysis across distributed data storage
- Great for processing highly parallel data
- Parallelizes the data processing as well

Map-Reduce Steps

- Map
  - At each place where relevant data lives do some initial processing
    - Basically whatever can be done without combining data
- Shuffle
  - By hashing map results, determine which processing server it should go to
  - Send it there
- Reduce
  - Combine received data to produce results

What’s Problematic about NoSQL?

- Transactions
  - Very little support
- Reports
  - Data correlation is a pain
- Queries
  - Generally written as computer programs
  - Changes are more expensive

NewSQL

- Have cake and eat it too?
- Well, not quite.
- What do we want?
  - Parallelization
  - Transactions
  - SQL support
    - Relational Model

NewSQL

- Distribute data across servers
- SQL Queries get automatically distributed
- Transaction support
  - But transactions get slow if they touch the same data a lot

NewSQL Architecture

- May be build on top of NoSQL Database engine
  - Add transaction support at expense of speed
- May be distributed storage engine for existing SQL database
- May be new architecture
  - Build to distribute both processing and storage
  - But with extra communication between nodes to handle transactions