

# Distributed Databases

CISC437/637, Lecture #21

Ben Carterette

Copyright © Ben Carterette

1

## Distributed Database Systems

- A **distributed database** consists of loosely coupled sites with no shared physical components
- A DBMS runs at each site, independent of the others
  - Contrast with a parallel DBMS: a single parent process that can use many CPUs
- Transactions may access data at one or more sites

Copyright © Ben Carterette

2

## Homogeneous vs Heterogeneous

- Distributed databases can be either **homogeneous** or **heterogeneous**
- Homogeneous:
  - Same DBMS software running at each site
  - Sites know about each other and agree to cooperate in processing requests
    - Sites give up control over software and schema
  - Appears as a single system to a user
- Heterogeneous:
  - Different DBMS software at each site
  - Possibly incompatible schema across sites
  - Sites may not be aware of each other

Copyright © Ben Carter@cs

3

## Classical View of Distributed DBs

- Two properties were considered desirable:
  - **Distributed data independence** – users should not have to know where the data is
  - **Distributed transaction atomicity** – users should be able to submit transactions that access data across sites just like purely local transactions
- In practice, these properties are not always easy to support, and may not even be desirable
- Distributed DBs is an evolving topic
  - The “right” way depends heavily on what the uses are

Copyright © Ben Carter@cs

4

## Distributed DB Architectures

- **Client-server**
  - DBMS servers manage data and execute transactions independently
  - Clients submit a query to a single server
- **Collaborating server**
  - DBMS servers cooperate in executing transactions spanning servers
  - One server receives a query; determines best distributed processing plan across server
- **Middleware systems**
  - One server is capable of distributing queries; the rest are completely local

Copyright © Ben Carter@Ge

5

## Distributing Data

- Relations will be stored on disks across sites
- Relations can be **fragmented** (partitioned) in different ways, but not re-partitioned during querying
  - **Horizontal fragmentation**: full rows stored at different sites
  - **Vertical fragmentation**: full columns stored at different sites
- Relations can be **replicated** at more than one site
  - For increased availability of data and faster query evaluation
  - **Synchronous vs asynchronous** determines how current the replicants are

Copyright © Ben Carter@Ge

6