Module II

Distributed Database Concepts

Introduction

- Distributed computing system
  - Consists of several processing sites or nodes interconnected by a computer network
  - Nodes cooperate in performing certain tasks
  - Partitions large task into smaller tasks for efficient solving
- Big data technologies
  - Combine distributed and database technologies
  - Deal with mining vast amounts of data

23.1 Distributed Database Concepts

- Distributed Database
  - A collection of multiple logically interrelated databases distributed over computer network, and a distributed DBMS as a software system that manages a distributed database
- What constitutes a distributed database?
  - Connection of database nodes over computer network
  - Logical interrelation of the connected databases
  - Possible absence of homogeneity among connected nodes
- Distributed database management system (DDBMS)
  - Software system that manages a distributed database

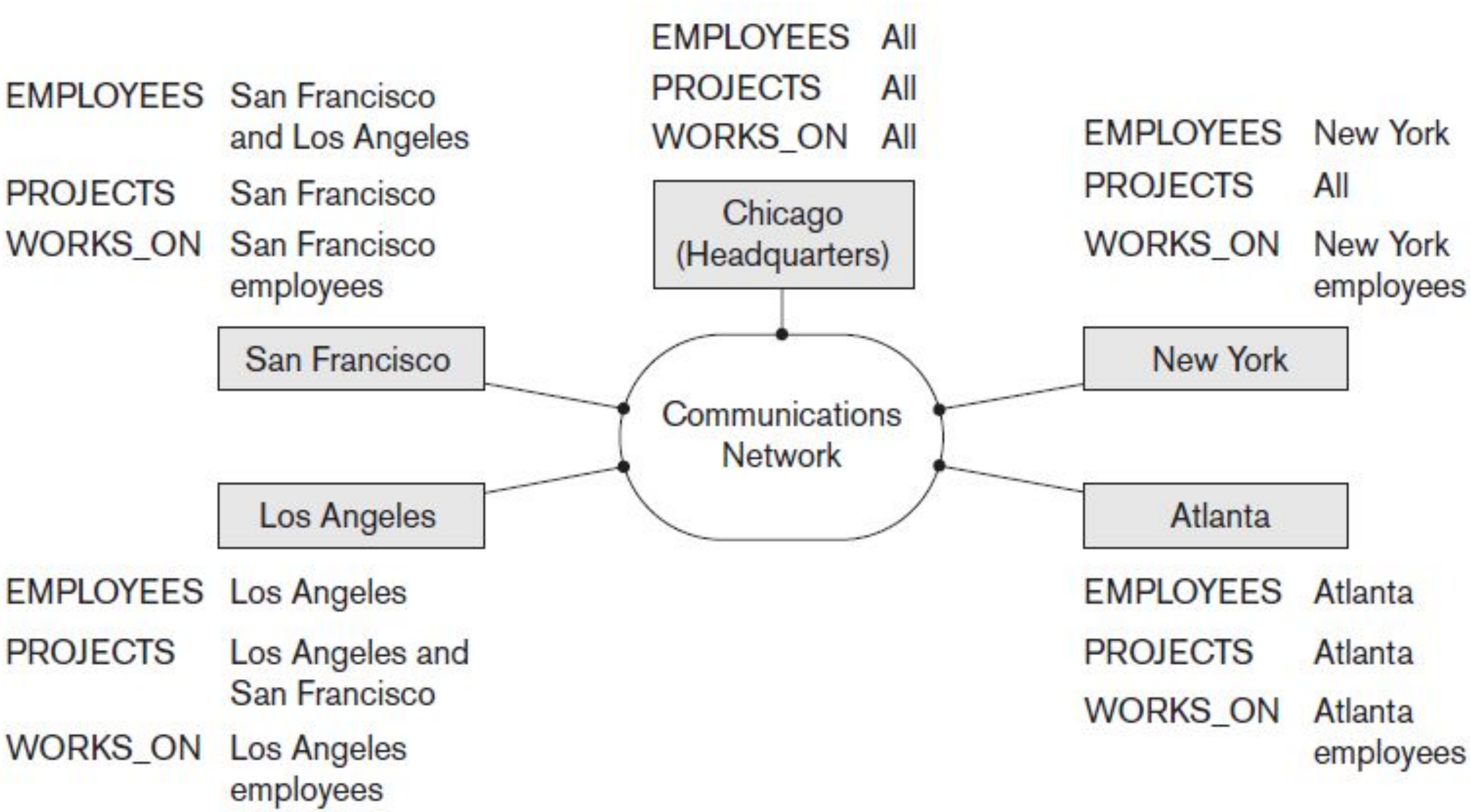
Distributed Database Concepts (cont'd.)

- Local area network
  - Hubs or cables connect sites
- Long-haul or wide area network
  - Telephone lines, cables, wireless, or satellite connections
- Network topology defines communication path
- Transparency
  - Hiding implementation details from the end user

Transparency

- Types of transparency
  - Data organization transparency
    - Location transparency
    - Naming transparency
  - Replication transparency
  - Fragmentation transparency
    - Horizontal fragmentation (sharding)
    - Vertical fragmentation
  - Design transparency
  - Execution transparency

Distributed Databases



Chicago (Headquarters)

EMPLOYEES	All
PROJECTS	All
WORKS_ON	All

San Francisco

EMPLOYEES	San Francisco and Los Angeles
PROJECTS	San Francisco
WORKS_ON	San Francisco employees

New York

EMPLOYEES	New York
PROJECTS	All
WORKS_ON	New York employees

Los Angeles

EMPLOYEES	Los Angeles
PROJECTS	Los Angeles and San Francisco
WORKS_ON	Los Angeles employees

Atlanta

EMPLOYEES	Atlanta
PROJECTS	Atlanta
WORKS_ON	Atlanta employees

Figure 23.1 Data distribution and replication among distributed databases

Availability and Reliability

- Availability
  - Probability that the system is continuously available during a time interval
- Reliability
  - Probability that the system is running (not down) at a certain time point
- Both directly related to faults, errors, and failures
- Fault-tolerant approaches

Scalability and Partition Tolerance

- Horizontal scalability
  - Expanding the number of nodes in a distributed system
- Vertical scalability
  - Expanding capacity of the individual nodes
- Partition tolerance
  - System should have the capacity to continue operating while the network is partitioned

Autonomy

- Determines extent to which individual nodes can operate independently
- Design autonomy
  - Independence of data model usage and transaction management techniques among nodes
- Communication autonomy
  - Determines the extent to which each node can decide on sharing information with other nodes
- Execution autonomy
  - Independence of users to act as they please

Advantages of Distributed Databases

- Improved ease and flexibility of application development
  - Development at geographically dispersed sites
- Increased availability
  - Isolate faults to their site of origin
- Improved performance
  - Data localization
- Easier expansion via scalability
  - Easier than in non-distributed systems

23.2 Data Fragmentation, Replication, and Allocation Techniques for Distributed Database Design

- Fragments
  - Logical units of the database
- Horizontal fragmentation (sharding)
  - Horizontal fragment or shard of a relation is a subset of the tuples in that relation
  - Can be specified by condition on one or more attributes or by some other method
  - Groups rows to create subsets of tuples
    - Each subset has a certain logical meaning

Data Fragmentation (cont'd.)

- Vertical fragmentation
  - Divides a relation vertically by columns
  - Keeps only certain attributes of the relation
- Complete horizontal fragmentation
  - Apply UNION operation to the fragments to reconstruct relation
- Complete vertical fragmentation
  - Apply OUTER UNION or FULL OUTER JOIN operation to reconstruct relation

Data Fragmentation (cont'd.)

- Mixed (hybrid) fragmentation
  - Combination of horizontal and vertical fragmentations
- Fragmentation schema
  - Defines a set of fragments that includes all attributes and tuples in the database
- Allocation schema
  - Describes the allocation of fragments to nodes of the DDBS

Data Replication and Allocation

- Fully replicated distributed database
  - Replication of whole database at every site in distributed system
  - Improves availability remarkably
  - Update operations can be slow
- Nonredundant allocation (no replication)
  - Each fragment is stored at exactly one site

Data Replication and Allocation (cont'd.)

- Partial replication
  - Some fragments are replicated and others are not
  - Defined by replication schema
- Data allocation (data distribution)
  - Each fragment assigned to a particular site in the distributed system
  - Choices depend on performance and availability goals of the system

Example of Fragmentation, Allocation, and Replication

- Company with three computer sites
  - One for each department
  - Expect frequent access by employees working in the department and projects controlled by that department
- See Figures 23.2 and 23.3 in the text for example fragmentation among the three sites

Types of Distributed Database Systems (cont'd.)

- Communication autonomy
  - Decide whether to communicate with another component DBS
- Execution autonomy
  - Execute local operations without interference from external operations by other component DBSs
  - Ability to decide order of execution
- Association autonomy
  - Decide whether and how much to share its functionality and resources