Distributed Databases and Client-Server Architectures

Chapter 25
Chapter Outline

- Distributed Database (DDB) Concepts
- Data Fragmentation, Replication and Allocation
- Types of Distributed Database Systems
- Query Processing
- Concurrency Control and Recovery
- 3-Tier Client-Server Architecture
- Distributed Database (DDB) Concepts …

- It is a system to process Unit of execution (a transaction) in a distributed manner. That is, a transaction can be executed by multiple networked computers in a unified manner.

- DDB can be defined as a collection of multiple logically related database distributed over a computer network.

- A distributed database management system (DDBMS) is a software system that manages a DDB while making the distribution transparent to the user.
Communications network

Site 1

Site 2

Site 3

Site 4

Site 5
- Advantages of DDB System

1. Management of distributed data with different levels of transparency
   - Distribution and Network transparency: (naming & location)
   - Replication transparency
   - Fragmentation transparency

2. Increased reliability and availability

3. Improved performance

4. Easier expansion (scalability)
- Data Fragmentation and Replication …

- Data Fragmentation
  - Horizontal fragmentation
  - Vertical fragmentation
  - Hybrid (Mixed) fragmentation

- Replication
  - Full
  - Partial
... - Data Fragmentation and Replication: e.g.

Communications network

- EMPLOYEES - All
- PROJECTS - All
- WORKS_ON - All

- EMPLOYEES - New York
  - PROJECTS - All
  - WORKS_ON - New York Employees

- EMPLOYEES - San Francisco and LA
  - PROJECTS - San Francisco
  - WORKS_ON - San Francisco Employees

- EMPLOYEES - LA
  - PROJECTS - LA and San Francisco
  - WORKS_ON - LA Employees

- EMPLOYEES - Atlanta
  - PROJECTS - Atlanta
  - WORKS_ON - Atlanta Employees
- Types of DDBs

- **Homogeneous**: Each site runs the same database system

- **Heterogeneous**: Each site may run different database system
  
  - **Federated**: access is managed through a single conceptual schema. This implies that the degree of local autonomy is minimum. Each site must adhere to a centralized access policy. There may be a global schema.

- **Multidatabase**: There is no one conceptual global schema. For data access a schema is constructed dynamically as needed by the application software.
-- Federated DB Management Systems Issues

- **Differences in data models:** Relational, Objected oriented, hierarchical, network, etc.

- **Differences in constraints:** Each site may have their own data accessing and processing constraints.

- **Differences in query language:** Some site may use SQL, some may use SQL-89, some may use SQL-92, and so on.
-- Homogeneous

Communications network

- Site 1 (Unix)
- Site 2 (Linux)
- Site 3 (Linux)
- Site 4 (Window)
- Site 5 (Window)

Oracle databases at each site.
-- Heterogeneous

Communications network

Object Oriented

Site 5

Unix

Relational

Site 1

Unix

Hierarchical

Window

Site 4

Object Oriented

Network DBMS

Site 3

Linux

Site 2

Linux

Relational
- Query Processing in DDBs ...

**Emp:**

<table>
<thead>
<tr>
<th>Fname</th>
<th>Minit</th>
<th>Lname</th>
<th>SSN</th>
<th>Bdate</th>
<th>Address</th>
<th>Sex</th>
<th>Salary</th>
<th>Superssn</th>
<th>Dno</th>
</tr>
</thead>
</table>

- Rows: 10,000
- Row size: 100 bytes.
- Table size: 1,000,000 bytes

**Dept:**

<table>
<thead>
<tr>
<th>Dname</th>
<th>Dnumber</th>
<th>Mgrssn</th>
<th>Mgrstartdate</th>
</tr>
</thead>
</table>

- Rows: 100
- Row size: 35 bytes.
- Table size: 3500 bytes
Assumption: cost of transferring data high: optimization necessary.

Example:

```
SELECT Fname, Lname, Dname
FROM Dept, emp
WHERE MgrSSN = SSN
```
Strategies

1. Transfer Emp and Dept to the result site and perform the join at site 3. Total bytes transferred = 1,000,000 + 3500 = \textbf{1,003,500 bytes}.

2. Transfer Emp to site 2, execute join at site 2 and send the result to site 3. Query result size = 40 * 100 = 4000 bytes. Total transfer size = 4000 + 1,000,000 = \textbf{1,004,000 bytes}. (Assume 40 is the size of the output record)

3. Transfer Dept relation to site 1, execute join at site 1 and send the result to site 3. Total transfer size = 4000 + 3500 = \textbf{7500 bytes}
Use **Semijoin** to reduce the number of tuples in a relation before transferring it to another site.

**Example:** Assume we want the results at site 2:

1. Project the join attributes of Dept at site 2, and transfer them to site 1. $9 \times 100 = 900$ bytes are transferred.

2. Join the transferred file with Emp at site 1, and transfer the required attributes from the resulting file to site 2. $39 \times 100 = 3900$ bytes are transferred.

3. Execute the query by joining the transferred file with Department and present the result to the user at site 2.
- Concurrency Control and Recovery in DDBs

- Distributed Databases encounter a number of concurrency control and recovery problems which are not present in centralized databases. Some of them are listed below.

- Dealing with multiple copies of data items
- Failure of individual sites
- Communication link failure
- Distributed commit
- Distributed deadlock
-- Techniques of Distributed Concurrency Control

- Based on Distinguished copy
  - Primary site with no backup site
  - Primary site with backup site
  - Primary copy Technique

- Based on voting
--- Primary site with no backup site …
Primary site with no backup site

A single site is designated as a primary site which serves as a coordinator for transaction management. If all transactions follow two-phase locking policy at all sites, then serializability is guaranteed.

**Advantages:**
- An extension to the centralized two phase locking so implementation and management is simple.
- Data items are locked only at one site but they can be accessed at any site.

**Disadvantages:**
- Bottleneck problem
- If the primary site fails, the entire system is inaccessible
--- Primary site with backup site

- Behaves as a shadow of primary site.
  - All locking information must be copied to the backup site

- In case of primary site failure, backup site can act as primary site and a new backup site is chosen

- **Advantage**: Faster recover from a failed primary site.

- **Disadvantage**: Slow because both sites must be consistent.
--- Recovery from a coordinator failure

- Failure of primary site with no backup site
  - All active transactions at all sites are aborted and restarted.
  - A new coordinator is selected and transaction processing is initiated

- Failure of primary site with backup site
  - All active transactions are suspended
  - The backup site is designated as the primary site
  - A new backup site is selected
  - The new backup site is sent all the location information from the new primary site.
--- Primary Copy Technique

- Lock coordination is distributed among various sites having distinguished copies of different data.

- **Advantages:**
  - Since primary copies are distributed at various sites, a single site is not overloaded with locking and unlocking requests.
  - Failure of one site affects only transactions whose primary copy resides at that site.

- **Disadvantages:**
  - Identification of a primary copy is complex.
  - A distributed directory must be maintained, possibly at all sites.
-- Concurrency control based on voting

- There is no primary copy of coordinator
- Send lock request to sites that have data item.
- If majority of sites grant lock then the requesting transaction gets the data item.
- Locking information (grant or denied) is sent to all these sites.
- To avoid unacceptably long wait, a time-out period is defined. If the requesting transaction does not get any vote information then the transaction is aborted.
Overview of 3-Tier C-S Architecture …

- Full scale DDBMS which supports all mentioned functionalities have not been developed yet.

- So far DDB application are being developed in the context of client-server (C-S) architecture.

- The most common C-S architecture is the 3-tier. It consists of:
  1. Presentation layer
  2. Application layer
  3. Database server
Overview of 3-Tier C-S Architecture
-- Presentation Layer

- Provides user Interface and interacts with the user using
  - Forms
  - Web interfaces

- Common programming languages used at this layer are:
  - HTML
  - JAVA
  - JavaScript
  - PERL
  - VB

- When web interface is used, the communication with the application layer is done via the HTTP protocol.
-- Application Layer

- Also called business logic layer
- Programs the application logic
- Can also handle additional application functionality, such as security checks and identity verification.
- Interacts with one or more databases servers using:
  - ODBC
  - JDBC
  - SQL/CLI
  - And other DB access techniques
Database server

- Handles queries and update requests from the application layer, processes the requests and sends back the results
  - If DBMS is relation or Object relational, requests come as SQL
  - Results most of the time return as XML
-- Processing of a SQL queries goes as follows

- The user enters his request using web interface or forms in the presentation layer.

- The application server formulates a user query based on the input from the presentation layer, decomposes it into a number of independent site queries. Each site query is sent to appropriate database server.

- Each server processes its query and sends the result (most probably in XML format) to the application server.

- The application server combines the results of site queries and formats it into HTML or some other form and sends it to the presentation layer.
END