Distributed Databases

Distributed database management system

• A distributed database (DDB) is a collection of multiple, logically interrelated databases distributed over a computer network.

• A distributed database management system (DDBMS) governs the storage and processing of logically related data over interconnected computer systems in which both data and processing functions are distributed among several sites.
Evolution

• Centralized databases (1970’s)
• Business operations became more decentralized geographically.
• Customer demands and market needs favored a decentralized management style.
• The decentralization of management structure based on the decentralization of business units made decentralized multiple-access and multiple-location databases a necessity.

Advantages

• DDBMS Advantages
  – Data are located near the “greatest demand” site.
  – Faster data access
  – Faster data processing
  – Growth facilitation
  – Less danger of a single-point failure

• DDBMS Disadvantages
  – Complexity of management and control
  – Security
  – Design is more complex
  – Increased storage requirements
  – Cost compared to a client/server model
Distributed Process versus Distributed Database

- *Distributed processing* shares the database’s *processing* among two or more physically independent sites that are connected through a network.
- *Distributed database* stores a *related database* over two or more physically independent sites connected via a computer network.
Components of a distributed database

- Servers
- Workstations
- Networks HW/SW
- Transaction Processor $TP$
- Data Processor $DP$
The importance of transparency

- Distribution transparency
- Transaction transparency
- Performance transparency
- Heterogeneity transparency

### Distribution transparency

- Three Levels of Distribution Transparency
  - Fragmentation transparency
  - Location transparency
  - Local mapping transparency

- Supported by a common data dictionary
Transaction Transparency

• Integrity is maintained
  – Remote Requests
  – Remote Transactions
  – Distributed Requests
  – Distributed Transactions

• Two phase commit

Two phase commit

• DP transaction log

• Protocol
  – DO-UNDO-COMMIT
  – Write ahead

• Co-ordinator and subordinates
Two phased commit (cont)

Phase 1: Preparation

- The coordinator sends a PREPARE TO COMMIT message to all subordinates.
- The subordinates receive the message, write the transaction log using the write-ahead protocol, and send an acknowledgement message to the coordinator.
- The coordinator makes sure that all nodes are ready to commit, or it aborts the transaction.

Phase 2: The Final Commit

- The coordinator broadcasts a COMMIT message to all subordinates and waits for the replies.
- Each subordinate receives the COMMIT message then updates the database, using the DO protocol.
- The subordinates reply with a COMMITTED or NOT COMMITTED message to the coordinator.
Performance transparency

- Directly related to query optimization

- Goal is to reduce costs associated with a request.
  - I/O cost
  - Communications cost
  - Processing cost

Design of a distributed database

- Partitioning
- Replication
- Location

- Data fragmentation allows us to break a single object into two or more segments or fragments.
- Each fragment can be stored at any site over a computer network.
Fragmentation

- Fragmentation strategies
  - Horizontal fragmentation
  - Vertical fragmentation
  - Mixed fragmentation

Replication of Data

- Data replication refers to the storage of data copies at multiple sites served by a computer network.

- Replicated data are subject to the mutual consistency rule, which requires that all copies of data fragments be identical.
Location

- Data allocation describes the processing of deciding where to locate data.
  - Centralized
  - Partitioned
  - Replicated