Distributed Database System

Basic definition

- System consists of a collection of sites, connected via communication networks.
- Each site is a database system site in its own right.
- Data: centralized logically, but physically spread over network.
- To the user, a distributed system should look exactly like a non-distributed system.
- DDBMS: a new software component at each site.
Advantages

- Advantages
  - Distributed nature of applications.
  - Higher reliability, availability, scalability.
  - Better performance.

Complexity with distribution

- Communication.
  - Different topologies, different protocols, etc
- Consistency Control.
  - Data distribution and replication.
- Concurrency Control.
  - Distributed locking & transaction management (2PC).
- Recovery.
  - New types of failures, such as network failure.
- Query processing.
  - Query decomposition & update propagation.
Objectives

• Local autonomy.
• No reliance on a central site.
• Continuous operation.
• Location independence/transparency.
• Fragmentation independence/transparency.
• Replication independence/transparency.
• Distributed query processing
• Distributed transaction management
• Hardware independence/transparency.
• Operating system independence/transparency.
• Network independence/transparency
• DBMS independence/transparency.

Types of distributed databases

• Homogeneous distributed database system
  – All sites run the same DBMS software.
• Heterogeneous distributed database system
  (Multidatabase system)
  – Different DBMS software (or non-DBMS software).
  – Heterogeneity
    • Data model.
    • Query language.
    • etc
Different Architectures

- Client-Server
- Schema Integration
  - Tightly-coupled solution.
- Federation
  - Loosely-coupled solution
- Middleware

Data Fragmentation

- Horizontal fragmentation.
  - Divides a relation horizontally by rows.
  - Can be specified by a Selection operation in relational algebra.
- Vertical fragmentation.
  - Divides a relation vertically by columns.
  - Can be specified by a Projection operation in relational algebra.
- Completeness.
- Reconstruct the relation.
Data replication and allocation

• Fully replicated.
  – Whole database at each site.

• No replication (non-redundant allocation)
  – Each fragment is stored at exactly one site.
  – All fragments must be disjoint (except PK in vertical fragmentation).

• Partial replication.
  – The number of copies of each fragment can range from one up to the total number of sites.
  – Replication schema describes the replication fragments.

Query processing

• Data transfer cost.
  – Data transfer cost is usually an optimization criterion in choosing a distributed query execution strategy.
  – Examples
    • Relations:
      Site₁: Student(SSN, Name, …, DeptNo) (10,000 records, 100 bytes each)
      Site₂: Dept(No, DeptName, …, MSSN) (100 records, 35 bytes each)
    • Queries (Site₃):
      – P_{SSN, Name, deptName}(Join_{DeptNo = No}(Student, Department))
      – P_{DeptName, name, SSN}(Join_{MSSN = SSN}(Dept, Student))
Query processing

• SemiJoin$_{A = B}$ (R, S)
  – A and B are domain-compatible attributes of R and S.
  – It produces the same results as the relational algebra expression $P_{<R>}(\text{Join}_{A=B}(R, S))$.
  – It is typically implemented by first transferring $F = P_{<B>}(S)$ to the site where R resides, and then joining F with R.
  – Note that semijoin is not commutative, say, SemiJoin$_{A = B}$ (R, S) $\neq$ SemiJoin$_{A = B}$ (S, R)

Query processing

• Query and update decomposition
  – Because of full distribution, fragmentation, and replication transparency.
  – Problems:
    • Decomposing query based on fragmentation.
    • Choosing right replica.
    • Update propagation.
    • Maintaining consistency.
Consistency among copies

- Synchronous replication
  - Read-any write-all
- Asynchronous replication
  - Primary copy replication.
  - Capture.
    - log-based capture.
    - snapshot-based capture.
  - Apply

Concurrency control & recovery

- Problems
  - Dealing with multiple copies.
  - Failure of individual sites.
  - Failure of communication links.
  - Distributed deadlock.
- Two solutions.
  - Distinguished copy of a data item.
    - Primary site technique, primary site with backup site, and primary copy technique.
  - Voting
Transaction management

- Two-phase commit protocol.
  - Phase 1: prepare phase
    - The global coordinator (usually the initiating site) asks participants to prepare (to promise to commit or rollback the transaction, even if there is a failure).
  - Phase 2: commit phase
    - If all participants respond to the coordinator that they are prepared, the coordinator asks all sites to commit the transaction. If any participants cannot prepare, the coordinator asks all sites to rollback the transaction.

Database interoperability

- Database gateways.
  - A point to point solution.
  - Tasks of a gateway
    - Implementing protocols for the exchange of information.
    - Mapping between data types.
    - Mapping between SQL dialect.
    - Mapping between feedback information.
    - Dealing with a variety of semantic mismatch.
    - Serving as a participant in the 2pl commit protocol.
    - Ensuring the distributed locking.
Database interoperability

- Data access middleware (Mediators).
  - Examples
    - DataJoiner from IBM Corp.
    - Cohera from Cohera Inc.
    - InfoHub from Sybase Inc.
  - Architecture