CSE 4125: Distributed Database Systems
Chapter - 1

Distributed Databases: An overview.

Outline

- Aspects and definition of DDB.
- Examples of DDB and non-DDB.
- DDB vs. traditional DB.
- Necessity of DDB.
- DDBMS.
- Accesses to DDB.

Aspects of DDB

- Distribution:
  - Data are not resident at same site (computer/ processor).
- Logical correlation:
  - Tied together.

Above definition is too vague.

Examples

- DDB on a geographically dispersed network.
  - Multiple DB, multiple computers, different locations, connected via communication network, global and local app.
- DDB on a local network.
  - Multiple DB, multiple computers, but same location, connected via LAN, global and local app.
- Multiprocessor system.
  - Multiple DB, Multiple computers as front-end, one computer as back-end, same location, connected via LAN, no global/ local app.

Definition

A distributed database is a collection of data which are distributed over different computers of a computer network. Each site of the network has autonomous processing capability and can perform local applications. Each site also participates in the execution of at least one global application, which requires accessing data at several sites using communication subsystem.

DDB vs. Traditional DB

- Centralized Control:
  - Traditional: Database Admin (DBA).
  - Distributed: Hierarchical Responsibility:
    - Global / Local DBA
- Data Independence:
  - Traditional: Data is transparent to programmer.
  - Distributed: Programs are written as if the databases are not distributed.

DDB vs. Traditional DB (cont.)

- Reduction of Redundancy:
  - Traditional: Redundancy is not desired.
  - Distributed: Desired. Because more copies means more recoveries.
- Efficient Access:
  - Traditional: Complex physical structure.
    - Navigate at record level.
  - Distributed: Distributed access plan.
    - Not navigate at record level.

DDB vs. Traditional DB (cont.)

- Integrity, recovery and concurrency control:
  - Common issue/ problem in both types.
  - Solution: transaction management.

Transaction

- An atomic unit of execution.
- Sequence of operation.
- Either completely performed, or not performed at all.
- Example: Fund transfer.

Transaction (cont.)

- Integrity:
  - Assuring one state to another.
- Recovery:
  - Preserving states while failure.
- Concurrency:
  - Synchronization.

Necessity of DDB

- Organizational and economic reason.
  - If the organization is –
    - Decentralized
    - DDB fits more economically.
- Interconnection of existing DB.
  - If need to exchange data.
  - If global application is necessary.

Necessity of DDB (cont.)

- Incremental growth.
- Reduced communication overhead.
  - One advantage of DDB is : local application does not engage communication network. Workload is distributed.
- Performance consideration.
  - Parallel processing can be done in DDB.
- Reliability and availability.
  - Graceful degradation.

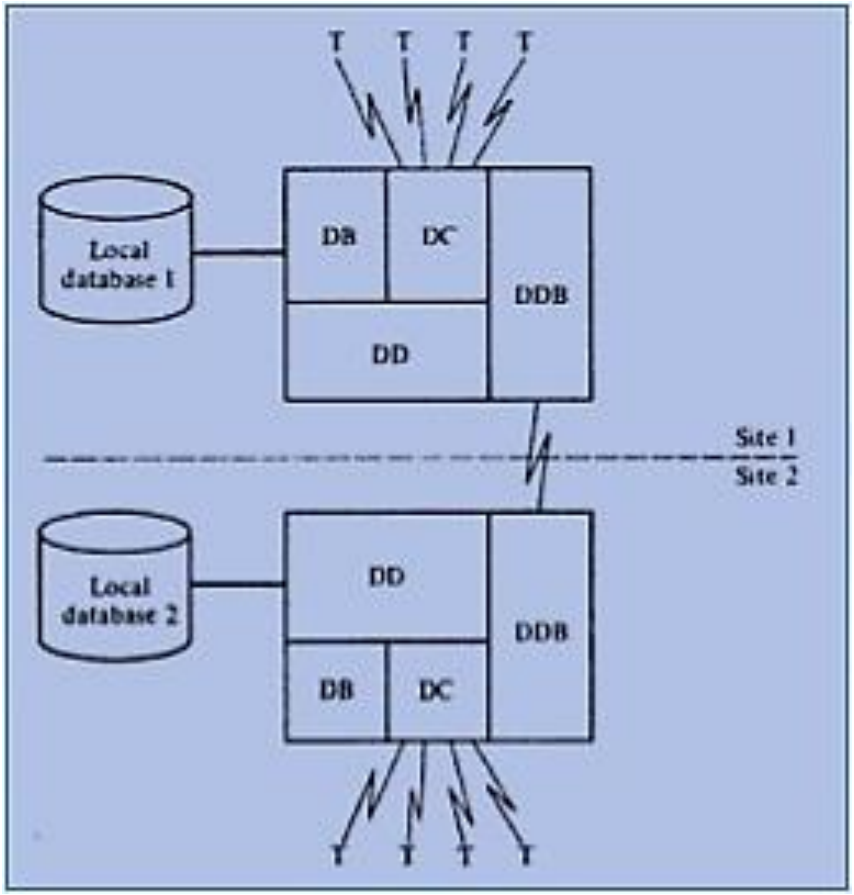
DDBMS

- Distributed Database Management System.
  - Extends the capabilities of DBMS by supporting communication between DBMSs.
  - DBMSs are installed at different sites (local DBMS).
- Supports creation and maintenance of DDB.

Components of DDBMS

- Database management component (DB).
- Data communication component (DC).
- Data dictionary (DD).
- Distributed database component(DDB).

Components of DDBMS (cont.)

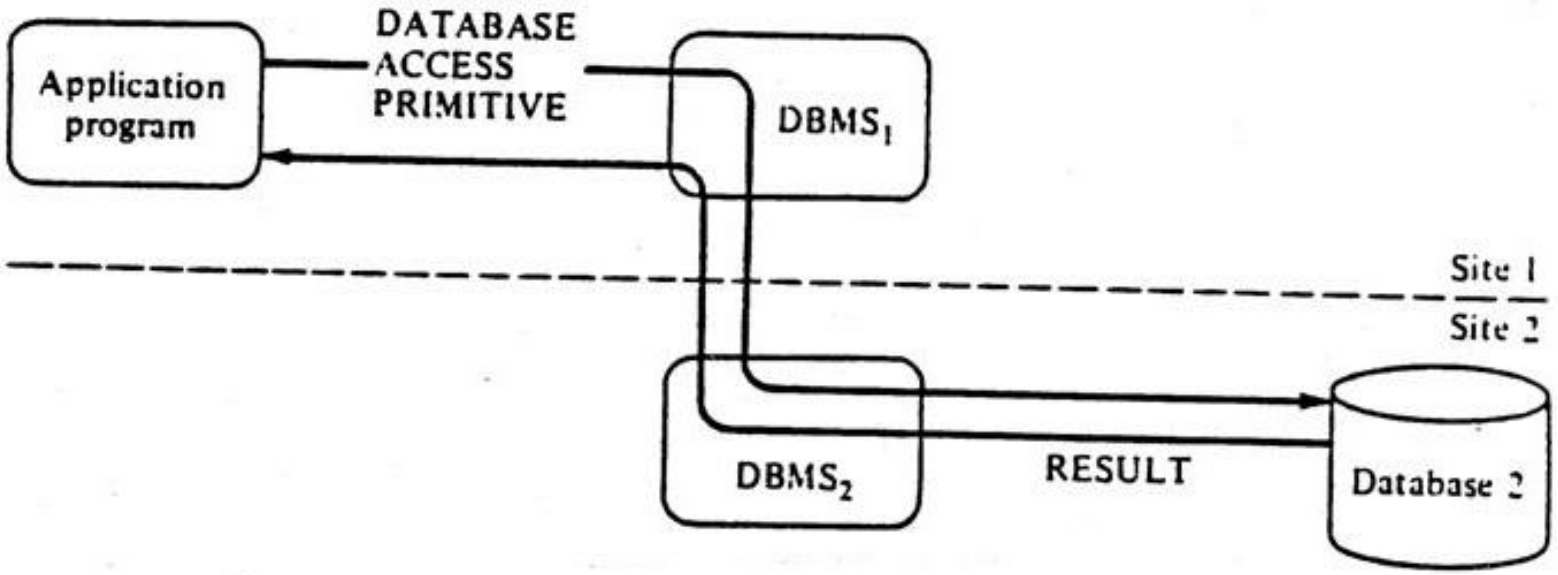


Types of DDBMS

- Homogeneous:
  - DDBMS with same DBMS at each site.
  - Ex. Oracle Database at every site.
- Heterogeneous:
  - DDBMS with at least two different DBMSs.
  - Ex. MS SQL Server, Oracle at different sites.
  - Need translating the different data models of the different local DBMSs.

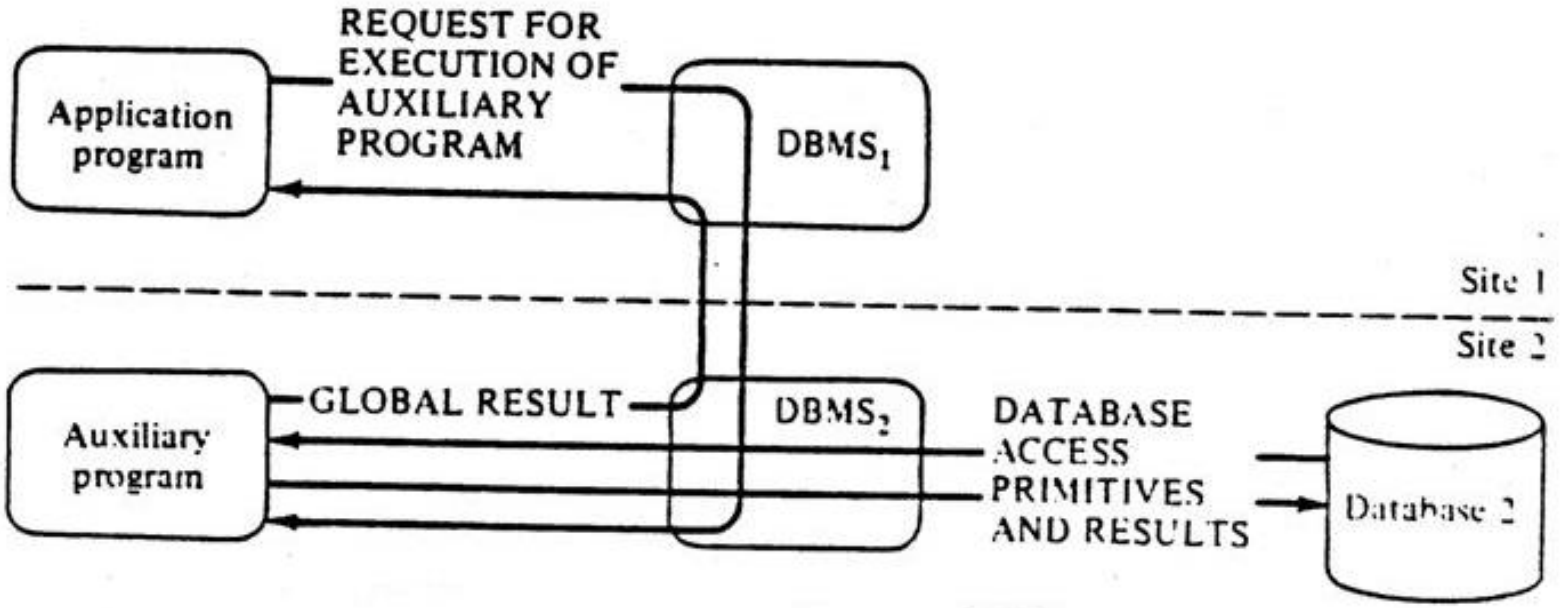
Accesses to DDB

- Remote access via DBMS primitives.



Accesses to DDB

- Remote access via an auxiliary program.



Sample question

1. Create your own scenario/ example of a distributed database over a local network (must be different from the ones shown in the class). Explain its necessity. Discuss its features.
2. Discuss the advantage and disadvantages of homogeneous and heterogeneous DDBMS.