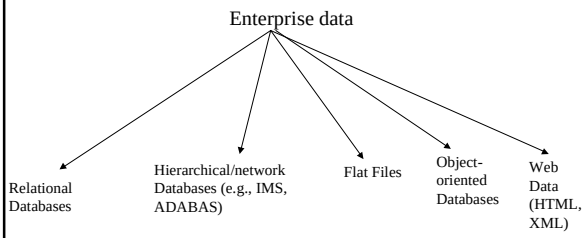
Enterprise Data and TransactionManagement

- Enterprise Data Management
- Processing XML Data
- Transaction Processing
- Distributed Transaction Processing
- Combining Data and Transaction management
- A Framework for Evaluating Data and Transaction management

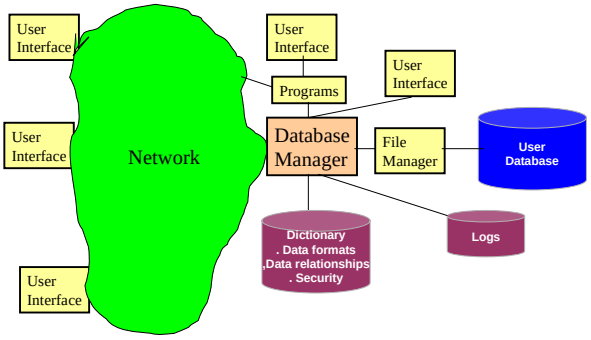
Amjad Umar

Enterprise Data



Enterprise Data: Enterprise Data (also known as Corporate Data) is the information that is used or created by a corporation in conducting business and is shared across the business processes of the corporation.

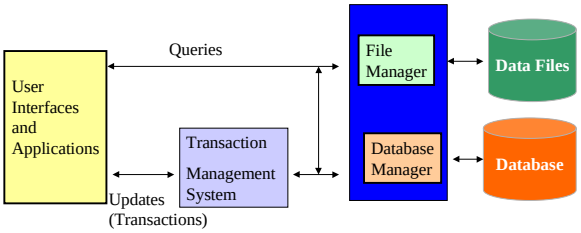
Database Management Systems



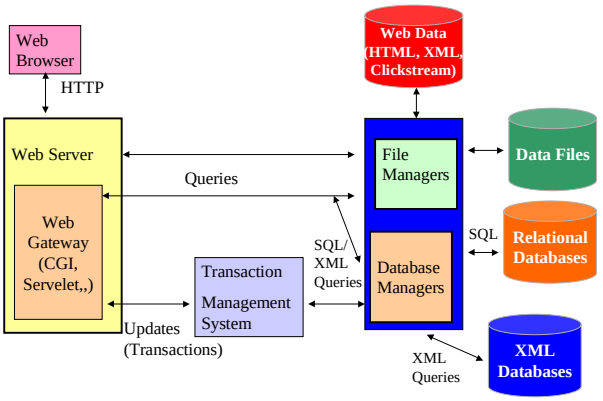
Enterprise Data in Distributed Environments

- Management of enterprise data in a distributed environment is a challenging task
  - Data at several sites in different formats
  - Several users need to access and update
  - How to design distributed databases
  - How to maintain data quality
  - How to use available technologies
- What is distributed data?
  - Different data at different computers (d1 at c1, d2 at c2)
  - Duplicate data- partial or complete (d1 at c1, c2, c3)
  - Partitioned (d1' at c1, d1'' at c2)

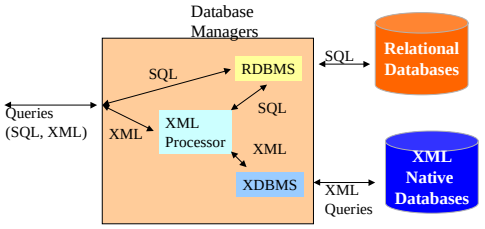
Conceptual View of Data and Transaction Management



Web-based Data and Transaction Management

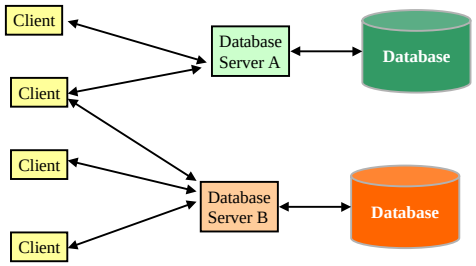


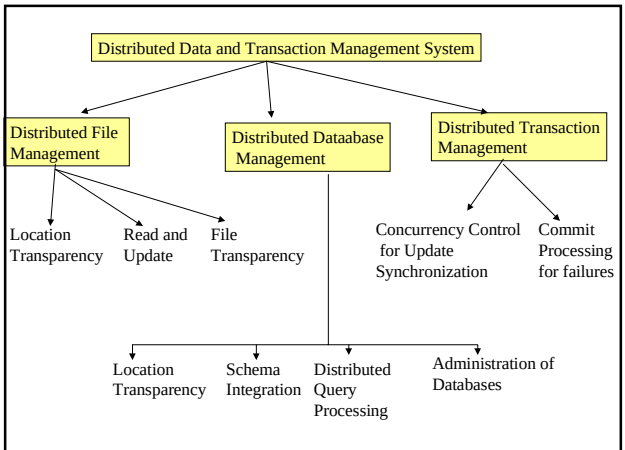
Mixing XML and SQL Databases



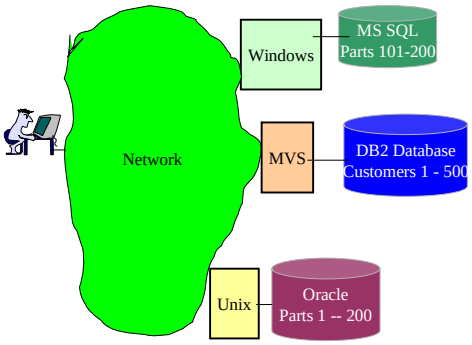
RDBMS = Relational Database Management System
XDBMS = XML Database Management System ("Native")

Single Site Remote Data Access

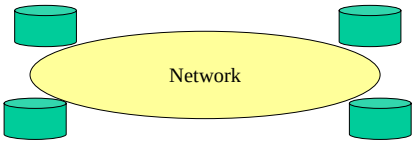




Database Management Systems



Distributed Data Management



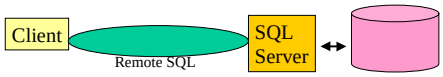
- Databases at different sites appear as one.
- User issues an SQL call: Select name, address from customers.
- The system finds the needed data.
- Distributed data management (DDM) supports transparency

  Retrieval transparency -

  Simple retrieval of ;information from a remote database is similar to local
  Remote joins: joins between multiple sites are supported

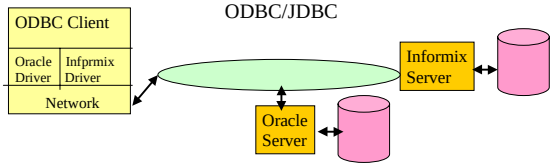
. Failure transparency - if one site fails, the data can be accessed from other sites

Simple Model: Remote SQL Middleware



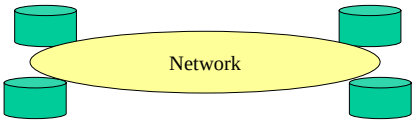
- Many SQL servers were developed for LANs.
- Mainframe database servers are now becoming available
- SQL server may be used to access non-SQL databases
  - A translator converts SQL to other formats
  - Example: EDA/SQL from Information Builders
- Categories of remote data access:
  - Basic: single server (retrieval, update)
  - Distributed query processing (retrieval from many servers, e.g., join between servers)
  - Database procedures: activate a precompiled (static) SQL procedure
  - Distributed transaction processing (update many servers in one call, e.g., distributed commits)

ODBC/JDBC



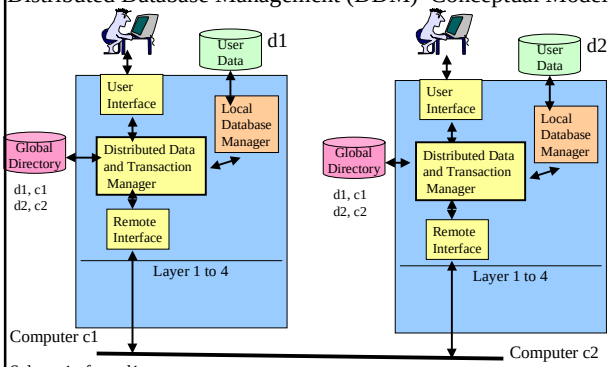
- Open Database Connectivity (ODBC) is an API introduced by Microsoft
- ODBC API is a Call Level Interface (CLI) not an embedded SQL interface
- ODBC allows a single application to access remote databases under different DBMSs (e.g., Informix, Oracle, Sybase).
- The same ODBC calls are used to access relational databases of different vendors.
- ODBC relies on data drivers to convert the ODBC calls to different database formats.
  - Informix data drivers needed to access Informix databases
  - Oracle data drivers needed to access Oracle databases
- A programmer can create an ODBC application without knowing the target DBMS.

Distributed Data and Transaction Management



- Databases at different sites appear as one.
- User issues an SQL call: Select name, address from customers.
- The system finds the needed data.
- Distributed data and transaction management (DDTM)
  1. Retrieval transparency - simple as well as joins
  2. Update transparency - data updated at many sites automatically
  3. Transaction transparency - transaction may be decomposed and executed at many sites
  4. Failure transparency - if one site fails, the data can be accessed from other sites

Distributed Database Management (DDM) Conceptual Model



Select a1,, from d1
Select a1,, from d2
Select a1,, from d1,d2
Update d2

Technical Challenges and Approaches

A. Database definition and translation
- Heterogeneous databases: IMS, DB2, OODBMS
- How to define data
- Where is data translated: originating/target
- Common approach: use SQL

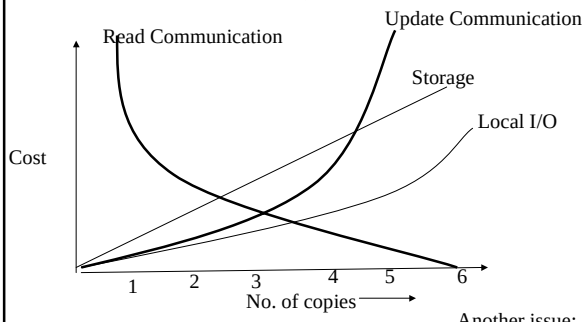
B. Directory allocation (name services)
- Directory shows what data is where (node)
- Directory at central site - communication
- Directory at every site - update
- Common approach: local data directory, go to regional for remote data ("differential directories")

Technical Challenges and Approaches (Cont.)

C. Database consistency/concurrency
- Consistency: correct reflection of reality
- Currency: recent information
- Concurrency: simultaneous access to shared data
- Update synchronization:

- Duplicate data - replicate changes
- Synchronization interval: transaction or interval
  - Two phase locking is used by centralized systems
- Lock, access, unlock
- Locking granularity: database, file, record, field
- Lock tables: where to locate them?
  - Deadlock (deadly embrace) is difficult to detect
  - Many database consistency control algorithms
  - Most commercially available DDBMS use some kind of locking

Technical Challenge: Data Allocation



•In practice, large number of copies are not practical
•Allocation of F files to N computers requires F. 2(N)
•Large number of decisions
•Reduce by reducing N and F

Another issue: Fault tolerance versus security

Transaction Processing


Customers

Deposits/withdrawals


Bank1

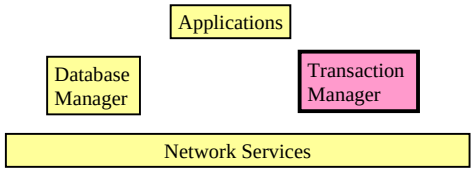
Electronic
Fund Transfer


Bank2

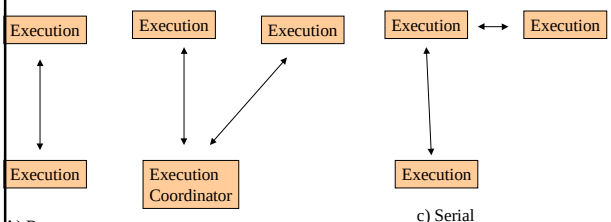
Many Electronic Commerce activities are transactions (purchasing, payment)

- Transaction: unit of consistency
  - "all or nothing at all" (everything complete, or go back)
  - Change of state (update) of something you care about
  - Example: Electronic fund transfer
- Transactions can be short duration (e.g., account update) or long (mortgage)
- Transaction managers: Responsible for correct execution of transaction
  - Handles murders and suicides
  - Roll back changes

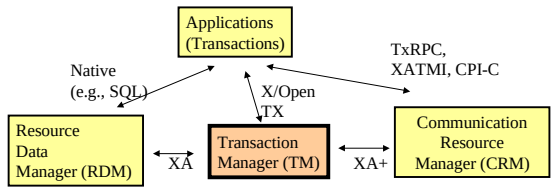
Transaction Managers (TP Monitors)



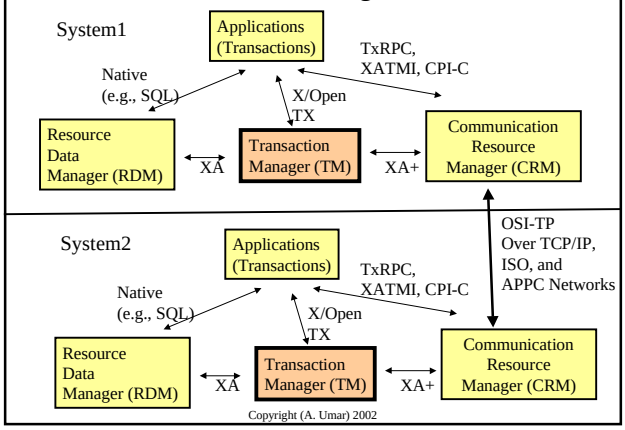
- Most database operations require transaction support
- TP middleware supports consistency and concurrency
- Traditional TP middleware on mainframes (CICS, IMS)
- Becoming available on C/S (Tuxedo, NCR TopEnd)
- Trend: transactions on CORBA



Standards



Transaction Processing Standards



Technical Challenges and Approaches (cont.)

D. Query optimization
  - Develop an access strategy to minimize communication (or response time)
- How to do the joins
- Where to access the data (if duplicate)
- Where to run the programs
  - Direct remote call (e.g. dynamic SQL)
  - Remote procedure call (e.g. static SQL)
  - Many query optimization algorithms
- For slow networks: minimize communication, use CPU
- For fast networks: maximize communication, avoid CPU
  - Solution approach for remote joins: semi-joins

Technical Challenges and Approaches

E. Failure detection and transaction recovery

_. When a transaction commits, updates are permanent

– If a transaction aborts, updates are rolled out

– In DDTMS, transaction may commit at one and abort at another
  - Update at node n1 completes
  - At n2 fails

– Common approach is two phase commit: 2PC ("TP Heavy")
  - Transaction indicates about to commit updates written to disk
  - Transaction says ok to commit updates are made permanent

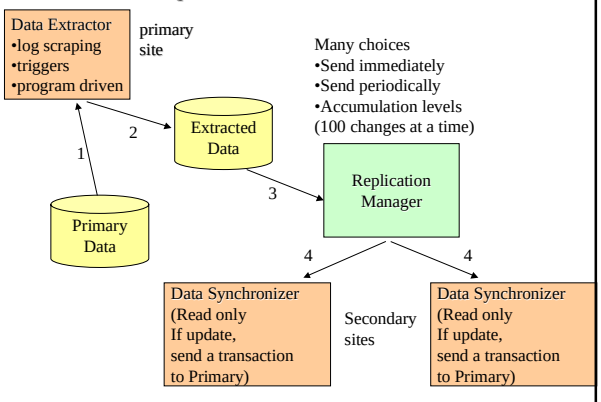
– Replication is becoming an alternative ("TP Lite")

-- Many vendors providing "Replication Servers"

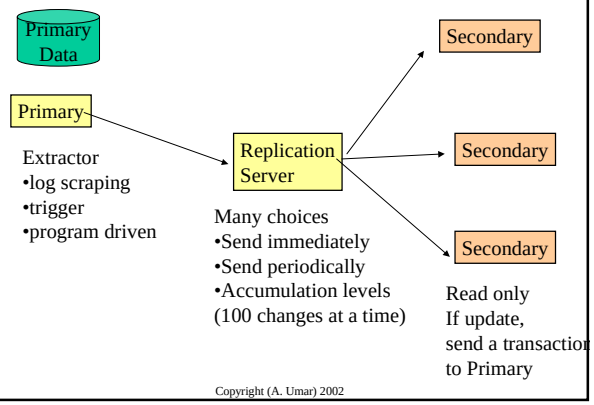
TP Heavy Versus Lite

- Two Approaches to maintain consistency:
  - TP Heavy: Two Phase Commit (immediate update)
    - All copies are updated (synchronized) within the duration of a transaction
    - If some update cannot be performed, rollback
  - TP Lite: Delayed update
    - Update one (master copy), others later
    - Use replicators for data synchronization (Oracle, Sybase)
    - Compensating transactions (after the fact cleanup)
- Synchronization interval: Time allowed for data to be inconsistent
- EC Transactions (e.g. buying from multiple suppliers) work well with TP lite approach

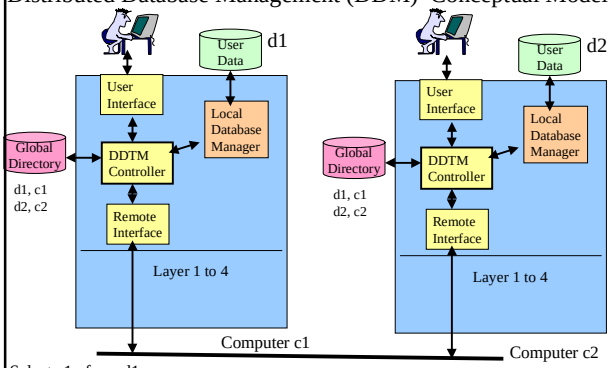
Data Replication Server Architecture



Replication Servers (General Model)

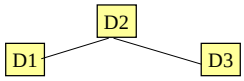


Distributed Database Management (DDM) Conceptual Model



Select a1,, from d1
Select a1,, from d2
Select a1,, from d1,d2
Update d2

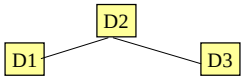
Evaluation Model



- Case 1: homogeneous, centrally controlled, no data duplication files uniquely assigned (partitioned) to nodes
  - A. Database definition and translation:
  - B. Directory allocation:
  - C. Database consistency/concurrency:
  - D. Query optimization:
  - E. Failure handling:
- Case 2: same as 1, copy can exist at host
  - A. Database definition and translation:
  - B. Directory allocation:
  - C. Database consistency/concurrency:
  - D. Query optimization:
  - E. Failure handling:

Evaluation Model

- Case 3: same as 2, several copies can exist
- A. Database definition and translation:
  - B. Directory allocation:
  - C. Database consistency/concurrency:
  - D. Query optimization:
  - E. Failure handling:
- Case 4: same as 3, not centrally controlled
  - A. Database definition and translation:
  - B. Directory allocation:
  - C. Database consistency/concurrency:
  - D. Query optimization:
  - E. Failure handling:
- Case 5: same as 4, heterogeneous
  - A. Database definition and translation:
  - B. Directory allocation:
  - C. Database consistency/concurrency:
  - D. Query optimization:
  - E. Failure handling:



Technology Assessment and Summary

- Database technologies are at the core of data warehousing and data minging
- Many technical challenges in handling distributed data
- Complexity increases with more copies of data
- Many vendors have products
- Almost all commercial DDTM support:
  - SQL
  - Single or two copies
  - Some restrictions on updating remote data
- TP Lite is quite popular:
  - Assign a primary among multiple copies
  - Primary copy is the only one updated
  - Primary copy updates are capyured and transmitted through replication servers
- DDTM technology is growing slowly.
- Replication gaining acceptance over 2PC.
- Most of the work surrounds SQL, but:
  - SQL conversion to non-SQL is not trivial
  - All SQLs are not same
- Standards can play a key role.

Selected References

- Date, C., “An Introduction to Database Systems”, Addison Wesley
- Elmasri and Navathe, “Fundamentals of Database Systems”, Benjamin Cummings
- Martin, D., “Advanced Database Techniques”, MIT Press
- Teorey, T.J., “Database Design”, Prentice Hall
- Umar, A., “Application (Re)Engineering: Building New and Dealing with Legacies”, Prentice Hall, Ch. 5
- Umar, A., “Object-Oriented Client-Server Internet Environments”, Prentice Hall, Ch. 5 and 6
- Magazines:
  - Data Management Review
  - Academic Conferences on VLDB (ACM), Data Engineering (IEEE)