# Outline

- Introduction
- Background
- Distributed DBMS Architecture
- Distributed Database Design
  - Fragmentation
  - Data Location
- Distributed Query Processing (Briefly)
- Distributed Transaction Management (Extensive)
- Building Distributed Database Systems (RAID)
- Mobile Database Systems
- Privacy, Trust, and Authentication
- Peer to Peer Systems

# Useful References

- W. W. Chu, *Optimal File Allocation in Multiple Computer System*, IEEE Transaction on Computers, 885-889, October 1969.
- Textbook *Principles of Distributed Database Systems*,

  Chapter 3.3, 3.4

# Allocation Alternatives

- Non-replicated
  - partitioned : each fragment resides at only one site
- Replicated
  - fully replicated : each fragment at each site
  - partially replicated : each fragment at some of the sites
- Rule of thumb:

If $\frac{\text{read - only queries}}{\text{update queries}} \geq 1$ replication is advantageous,

otherwise replication may cause problems

# Comparison of Replication Alternatives

| | Full-replication | Partial-replication | Partitioning |
|---|---|---|---|
| QUERY PROCESSING | Easy | ← Same Difficulty → | |
| DIRECTORY MANAGEMENT | Easy or Non-existant | ← Same Difficulty → | |
| CONCURRENCY CONTROL | Moderate | Difficult | Easy |
| RELIABILITY | Very high | High | Low |
| REALITY | Possible application | Realistic | Possible application |

# Information Requirements

- Four categories:
  - Database information
  - Application information
  - Communication network information
  - Computer system information

# Fragment Allocation

- Problem Statement

  Given

  $F = \{F_1, F_2, \ldots, F_n\}$ fragments

  $S = \{S_1, S_2, \ldots, S_m\}$ network sites

  $Q = \{q_1, q_2, \ldots, q_q\}$ applications

  Find the "optimal" distribution of $F$ to $S$.
- Optimality
  - Minimal cost
    - Communication + storage + processing (read & update)
    - Cost in terms of time (usually)
  - Performance

    Response time and/or throughput
  - Constraints
    - Per site constraints (storage & processing)

# Information Requirements

- Database information
  - selectivity of fragments
  - size of a fragment
- Application information
  - access types and numbers
  - access localities
- Communication network information
  - unit cost of storing data at a site
  - unit cost of processing at a site
- Computer system information
  - bandwidth
  - latency
  - communication overhead

# Allocation

File Allocation (FAP) vs Database Allocation (DAP):

- Fragments are not individual files
  - relationships have to be maintained
- Access to databases is more complicated
  - remote file access model not applicable
  - relationship between allocation and query processing
- Cost of integrity enforcement should be considered
- Cost of concurrency control should be considered

# Allocation – Information Requirements

- Database Information
  - selectivity of fragments
  - size of a fragment
- Application Information
  - number of read accesses of a query to a fragment
  - number of update accesses of query to a fragment
  - A  matrix indicating which queries updates which fragments
  - A similar matrix for retrievals
  - originating site of each query
- Site Information
  - unit cost of storing data at a site
  - unit cost of processing at a site
- Network Information
  - communication cost/frame between two sites
  - frame size

# Allocation Model

**General Form**

min(Total Cost)

subject to

response time constraint

storage constraint

processing constraint

**Decision Variable**

$$x_{ij} = \begin{cases} 1 \text{ if fragment } F_i \text{ is stored at site } S_j \\ 0 \text{ otherwise} \end{cases}$$

# Allocation Model

- Total Cost

$$\sum_{\text{all queries}} \text{query processing cost} +$$

$$\sum_{\text{all sites}} \sum_{\text{all fragments}} \text{cost of storing a fragment at a site}$$

- Storage Cost (of fragment $F_j$ at $S_k$)

  $(\text{unit storage cost at } S_k) * (\text{size of } F_j) * x_{jk}$

- Query Processing Cost (for one query)

  processing component + transmission component

# Allocation Model

- Query Processing Cost

  Processing component

  access cost + integrity enforcement cost + concurrency control cost

  - Access cost

  $$\sum_{\text{all sites}} \sum_{\text{all fragments}} (\text{no. of update accesses} + \text{no. of read accesses}) * x_{ij} * \text{local processing cost at a site}$$

  - Integrity enforcement and concurrency control costs
    - Can be similarly calculated

  

# Allocation Model

- Query Processing Cost

  Transmission component

  cost of processing updates + cost of processing retrievals

  - Cost of updates

    $$\sum_{\text{all sites}} \sum_{\text{all fragments}} \text{update message cost} \ +$$

    $$\sum_{\text{all sites}} \sum_{\text{all fragments}} \text{acknowledgment cost}$$

  - Retrieval Cost

    $$\sum_{\text{all fragments}} \min_{\text{all sites}}(\text{cost of retrieval command} \ + \text{cost of sending back the result})$$

# Allocation Model

- Constraints
  - Response Time

    execution time of query $\leq$ max. allowable response time for that query

  - Storage Constraint (for a site)

    $$\sum_{\text{all fragments}} \text{storage requirement of a fragment at that site} \leq \text{storage capacity at that site}$$

  - Processing constraint (for a site)

    $$\sum_{\text{all queries}} \text{processing load of a query at that site} \leq \text{processing capacity of that site}$$

# Allocation Model

- Solution Methods
  - FAP is NP-complete
  - DAP also NP-complete
- Heuristics based on
  - single commodity warehouse location (for FAP)
  - knapsack problem
  - branch and bound techniques
  - network flow

# Allocation Model

- Attempts to reduce the solution space
  - assume all candidate partitionings known; select the “best” partitioning
  - ignore replication at first
  - sliding window on fragments