Outline

- Introduction
- Background
- Distributed DBMS Architecture
  - Introduction to Database Concepts
  - Alternatives in Distributed Database Systems
  - Datalogical Architecture
  - Implementation Alternatives
  - Component Architecture
- Distributed Database Design (Briefly)
- 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

- Textbook *Principles of Distributed Database Systems*,
  Chapter 1.7
Alternatives in Distributed Database Systems

- Distribution
- Heterogeneity
- Autonomy

- Client/server
- Peer-to-peer
- Distributed DBMS
- Federated DBMS
- Distributed multi-DBMS
- Multi-DBMS
Dimensions of the Problem

- **Distribution**
  - Whether the components of the system are located on the same machine or not

- **Heterogeneity**
  - Various levels (hardware, communications, operating system)
  - DBMS important one
    - data model, query language, transaction management algorithms

- **Autonomy**
  - Not well understood and most troublesome
  - Various versions
    - **Design autonomy**: Ability of a component DBMS to decide on issues related to its own design.
    - **Communication autonomy**: Ability of a component DBMS to decide whether and how to communicate with other DBMSs.
    - **Execution autonomy**: Ability of a component DBMS to execute local operations in any manner it wants to.
Datalogical Distributed DBMS Architecture

ES\(_1\)  
\(\cdots\)  
ES\(_n\)  

GCS

LCS\(_1\)  
\(\cdots\)  
LCS\(_n\)

LIS\(_1\)  
\(\cdots\)  
LIS\(_n\)

ES: External Schema  
GCS: Global Conceptual Schema  
LCS: Local Conceptual Schema  
LIS: Local Internal Schema
Datalogical Multi-DBMS Architecture

GES\_1 \quad GES\_2 \quad \ldots \quad GES\_n

LES\_1\_1 \quad \ldots \quad LES\_1\_n \quad \ldots \quad \ldots \quad \ldots \quad LES\_n\_m

LCS\_1 \quad LCS\_2 \quad \ldots \quad LCS\_n

LIS\_1 \quad LIS\_2 \quad \ldots \quad LIS\_n

GES: Global External Schema
LES: Local External Schema
LCS: Local Conceptual Schema
LIS: Local Internal Schema
Timesharing Access to a Central Database

- No data storage
- Host running all software

Terminals or PC terminal emulators

Batch requests  Response

Network

Database

Communications

Application Software

DBMS Services
Multiple Clients/Single Server
## Task Distribution

<table>
<thead>
<tr>
<th>Application</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>QL Interface</td>
<td>Programmatic Interface</td>
</tr>
<tr>
<td>Communications Manager</td>
<td></td>
</tr>
</tbody>
</table>

- SQL query
- result table

**Communications Manager**

- Query Optimizer
- Lock Manager
- Storage Manager
- Page & Cache Manager

**Database**
Advantages of Client-Server Architectures

- More efficient division of labor
- Horizontal and vertical scaling of resources
- Better price/performance on client machines
- Ability to use familiar tools on client machines
- Client access to remote data (via standards)
- Full DBMS functionality provided to client workstations
- Overall better system price/performance
Problems With Multiple-Client/Single Server

- Server forms bottleneck
- Server forms single point of failure
- Database scaling difficult
Multiple Clients/Multiple Servers

- directory
- caching
- query decomposition
- commit protocols

Diagram:
- Applications
- Client Services
- Communications
- LAN
- Communications
- DBMS Services
- Database
- Communications
- DBMS Services
- Database
Server-to-Server

- SQL interface
- programmatic interface
- other application support environments

LAN

Database

Communications

DBMS Services

Applications

Client Services

Communications

Database
Components of a Multi-DBMS

Distributed DBMS

© 1998 M. Tamer Özsu & Patrick Valduriez
Directory Issues

<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global &amp; central</td>
<td>Local &amp; central</td>
</tr>
<tr>
<td>&amp; non-replicated</td>
<td>&amp; non-replicated</td>
</tr>
<tr>
<td>(?</td>
<td>(?</td>
</tr>
<tr>
<td>Local &amp; central</td>
<td>Global &amp; central</td>
</tr>
<tr>
<td>&amp; replicated (</td>
<td>&amp; non-replicated (?</td>
</tr>
<tr>
<td>Global &amp; distributed</td>
<td>Local &amp; distributed</td>
</tr>
<tr>
<td>&amp; non-replicated (</td>
<td>&amp; non-replicated</td>
</tr>
<tr>
<td>Local &amp; central</td>
<td>Global &amp; distributed</td>
</tr>
<tr>
<td>&amp; replicated (</td>
<td>&amp; non-replicated (?)</td>
</tr>
<tr>
<td>Global &amp; central</td>
<td>Local &amp; distributed</td>
</tr>
<tr>
<td>&amp; replicated (</td>
<td>&amp; replicated</td>
</tr>
<tr>
<td>Local &amp; central</td>
<td>Global &amp; distributed</td>
</tr>
<tr>
<td>&amp; replicated (?)</td>
<td>&amp; replicated (?)</td>
</tr>
<tr>
<td>Global &amp; central</td>
<td>Local &amp; distributed</td>
</tr>
<tr>
<td>&amp; replicated (?)</td>
<td>&amp; replicated</td>
</tr>
<tr>
<td>Local &amp; central</td>
<td>Global &amp; distributed</td>
</tr>
<tr>
<td>&amp; replicated (</td>
<td>&amp; replicated (?)</td>
</tr>
<tr>
<td>Global &amp; central</td>
<td>Local &amp; distributed</td>
</tr>
<tr>
<td>&amp; replicated (?)</td>
<td>&amp; replicated</td>
</tr>
</tbody>
</table>

Distributed DBMS

© 1998 M. Tamer Özsu & Patrick Valduriez