Chapter 4

Distributed DBMS Architecture
Introduction

• Architecture defines a system’s structure with
  ◦ Components
  ◦ Functions of components, and
  ◦ Their interactions

• Purpose of ‘reference architecture’:
  ◦ A framework for discussion
  ◦ Standardization
4.1 Transparencies in a distributed DBMS

- X transparency means the existence of X is not known to users.
- Closely related to architecture issues.
4.1.1 Data independence

- Same as in centralized DBMS:
  - Logical data independence
    - applications are immune to changes in schema
  - Physical independence
    - applications are immune to changes in physical storage, and data distribution
4.1.2 Network transparency
- Also known as “distribution transparency”

4.1.3 Replication transparency

4.1.4 Fragmentation transparency
- Data fragmentation – the division of relation into smaller fragments at different sites.
4.2 DBMS Standardization
4.2.1 ANSI/APARC Architecture

• A 3-level architecture based on data organization

<table>
<thead>
<tr>
<th>View</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>External View</td>
<td>• How users view the shared data</td>
</tr>
<tr>
<td></td>
<td>• Commands used: create view ... as ...</td>
</tr>
<tr>
<td>Conceptual View</td>
<td>• Synthesis of external views of an enterprise</td>
</tr>
<tr>
<td></td>
<td>• Commands used: create db ... create table ...</td>
</tr>
<tr>
<td>Internal View</td>
<td>• Physical definition and organization of data</td>
</tr>
<tr>
<td></td>
<td>• Commands used: index on device segment definition and allocation</td>
</tr>
</tbody>
</table>
The ASNI/SPARC Architecture (1997)
4.3 Architecture Models for Distributed DDBMS
4.3 Architecture Models for Distributed DDBMS

- Classification of multiple DBs shared by multiple DBMSs

- Criteria for classification
  - **Distribution**
    - 0: centralized, 1: C/S, 2: P2P
  - **Autonomy**
    - 0: tight integration, 1: semi-autonomous, 2: total isolation
  - **Heterogeneity**
    - 0: homogeneous, 1: heterogeneous
1. (A0,D2,H0)
P2P homogeneous tightly integrated

2. P2P heterogeneous federated

3. (A2,D2,H1)
P2P heterogeneous multidatabase

Note 1: the available most advanced commercial product
2 and 3: active research area
Autonomy: the distribution of control

- (0) tightly integrated
  - a single image of the entire DB to and user and a single DBMS

- (1) semi-autonomous
  - all DBMSs can operated independently but have to join a federation to make local data sharable

- (2) total isolation
  - each DBMS is stand-alone and enjoy 100% freedom of autonomy
Distinction between DDB and MDB

- In MDB
  - every local data manager is a complete DBMS with its own DDL, DML and TM

- In DDB
  - there is only one data manager with its dependent component-data-manager at every local site

The focus of this course is the PRINCIPLES of DDBs (A0, D2, H0)
4.3.1 Distributed DBMS architecture

- An architecture based on data organization

![Diagram showing the distributed DBMS architecture]

- External Schema (ES)
- Global Conceptual Schema (GCS)
- Local Conceptual Schema (LCS)
- Local Internal Schema (LIS)
Detailed functional description of DDBMS

• Legend
  - Administrative Role
  - Data Dictionary/Directory
  - Processing Unit
Components of a Distributed DBMS

Semantic data controller
- Integrity constraint and authorization

Global Query Processor
- decompose and optimize global query

Global Execution Monitor
- Distributed transaction manager

Runtime Support Processor
- physical access to database
Client/Server Systems

- A computing model dividing functions between client and server
Discussion

- Why DDB?
- Compare DDB with CDB, MDB, C/S DB.
- Do you understand DDB architecture?