Introduction to Distributed Databases

I. Basic Concepts and Notions

II. Distributed DB Design

III. Distributed Query Processing

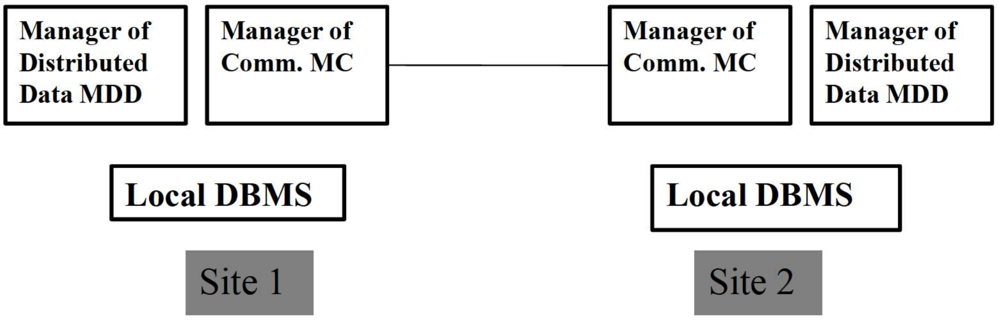
PSU - Master - Hameurlain@irit.fr

I.1. Basic Concepts and Notions

- An Example of a Distributed Database
- Motivations
  - Data sharing of a company located at several sites
  - Development of Distributed Systems
  - Data Sharing : (i) Interconnection network of computers and (ii) Specialized systems to manage distributed data

- Definition: Distributed Database
  - « A set of cooperating databases, each located/residing at a different site, that the user views and manipulates as a centralized database»
    - ➔ The distribution is *transparent* to the user

I.2. Management of Distributed DB



- **Distributed DBMS:** The management of DDB requires the following system components at each site :
  - {Manager of Dist. Data, Manager of Comm., Local DBMS}

I.3. Objectives of the Distributed DB (1/3)

Location Independence/Transparency

- Users can ignore the data location
- Transparency provides physical independence from the dist. Env.
- Data location information is maintained in the data dictionary and used by the DDBMS to find data

➔ « The main advantage is that the DB may be physically reorganized by moving relations to different sites, without any impact on the application programs that access them »

Fragmentation Transparency

- Fragmentation: Divide a relation into sub-relations (called fragments) :
- Fragment definition: by restriction op. (Horiz. Frag.) and/or projection (Vertical Frag.).
- Fragmentation improves the performance of DB queries by increasing the **data locality principle** : **"The data should be located where they are most often accessed"**
- An example: EMP (eno, name, location) ➔ EMP1/Paris & EMP2/London
- Queries on logical relations (*Global Queries*) are mapped by DDBMS (using Data Dictionary DD) into queries on fragments (*Fragment Queries*).

I.3. Objectives of the Distributed DB (2/3)

■ Independence to replication (Replication Transparency)

- A fragment is replicated when it is stored as 2 or more copies, each at a different site
- Objectives : (i) Data Availability (complexes queries, Fault Tolerant/reliability), (ii) High Performance (by increasing and exploiting the principle of data locality).
- Advantages of replication should be compared to the cost of maintaining copies in the case of updates (Cost required to keep the copies identical), because "The update to one copy must be propagated to all its copies"
- Replication information is maintained in the data dictionary and used by the DDBMS to find the relevant copy (Distributed Query Processing)

■ Independence to DBMS (DBMS Transparency)

- Hide the fact that local DBMSs may be different (Models and languages).
- That is the objective of Heterogeneous Distributed DB

➔ Mapping/Translator of a Data Model and its Language for each DBMS

➔ Choice of a Common/Pivot Model and its Language (e.g. Relational Model and declarative Language (SQL)).

I.3. Objectives of the Distributed DB (3/3)

Site Autonomy:

- Each site must store (locally) all Data Dictionary Information.
- This allows to each site to control and manipulate local data independently/regardless of others sites

  ➔ The Administration of a DDB can decentralized

Expandability of DDB

- Adding new site(s) with minimal impact on the Appli. Programs

  ➔ This allows to improve processing capability and storage capacity

Increased Performance

<Minimizing the Response Time, Maximizing the System Throughput >

"Can be achieved by carefully fragmenting the data and placing the data (data replication) which promote parallelism when processing a distributed query".

I.4. Distributed DBMS functions

1. Management of a Global Data Dictionary
2. Definition of Distributed Data
3. Semantic Control of Distributed Data

   <View management, Data Security/Authorization Control, Semantic Control Integrity>
4. Processing of Distributed Queries
5. Transaction Management

I.5. Reference Architecture for DDB (1/3)

Global Schema GS

- Emp (enu, name, title, salary, *dnu*), Dept(dnu, location, …)
- GS defines all the relations contained in the DDB.

  ➔ Provides physical data independence from the distributed environment

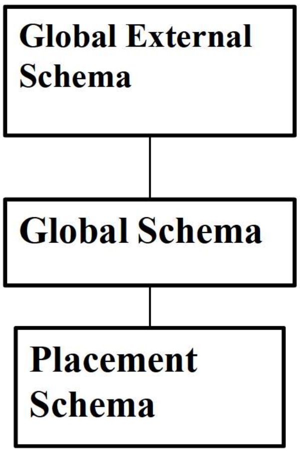
  ➔ Logical data independence is achieved by the global external schemas

I.5 Reference Architecture for DDB (2/3)

- Global External Schema GES
  - View of the DDB for a user class

    ➔ Logical Independence
- Placement Schema

  Indicates the way the relations are Placed in the network
  - Contains all info. regarding data fragmentation, replication and location
  - + transfo. Rules of the global schema data into localized, fragmented and duplicated data (see Data Localization Process in Dist. Quer. Proc.)

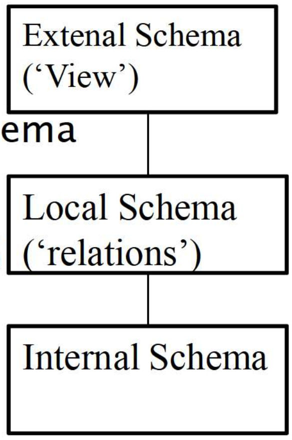


I.5. Reference Architecture for DDB (3/3)

- Local Schema LS (at site level)
  - External Schema

  Frag that will be described in placement schema

- Distributed/Global Query
  - Query expressed/formulated on GS
- Local Query
  - Query expressed/formulated on LS



I.6. Fonctional Architecture of a Dist DBMS (1/2)

- User Interface (see Figure 12.4, p. 427, Chapter 12, DDB)
- Semantic Control
  - View Query ∈ Global External Schema ⇒ Req Global Query (Rel) ∈ GS
    - Autorization CTRL+ Semantic Integrity CTRL
- Distributed Query Processing
  - Req Rel ∈ SG ➔ *Distributed Execution Plan DEP*
    - Optimized sequence of local queries with synchronization and communications operators (Inter-site exchange of intermediate data).
- Transaction Management: coordinates the execution of a distributed query, by communicating with manager of the site data, on which local queries are executed as multiple sub-transactions.

I.6. Fonctional Architecture of a Dist DBMS (2/2)

Local Query Processing

- Query expressed according to the placement schema ➔ Query expressed on the local external schema
  - Translation of a pivot language ➔ Language of the local DBMS
- **Next Step:** report to Uniprocessor or Parallel DBMS

I.7. Data Dictionary (1/2)

- DD Roles
  - Contains info on External GS, GS and placement schema
  - Used to translate, analyze and optimize queries
  - Managed as a Dist. DB: Fragmented and replicated
  - Placement of DD : (i) degree of autonomy desired for each site, and (ii) Expected Performance
- Organization of DD: 3 Approaches
  - Centralized
  - Replicated
  - Distributed

I.7. Data Dictionary (2/2)

- Organization of DD: 3 Approaches
  - Centralized
    - + Single site : Administration and updating are easy and efficient
    - – Remote Access by other queries/sites
    - – If the server/site breaks down ➔ Dist. DB is unusable
  - Replicated
    - – Manage a copy of DD on each site
    - + Decomposition & optimization are efficient
    - – Pb of consistency if the DD is updated
  - Distributed
    - + Principle of Data Locality (Local data Manipulation is efficient)
    - – Remote Data Manipulation is more complicated : (i) detection of no local data, and (ii) determination of their definition site