



جامعة طرابلس

كلية تقنية المعلومات

قواعد البيانات النقالة والغير متجانسة

Heterogeneous and Mobile Databases

ITMC322

أستاذ المادة / محمد أوهيبة

المحاضرة الخامسة

Review of topics

- ***Distributed Database***

- ***Introduction***

- ***Data distribution***

- ***Data Fragmentation***

- ***Data Replication***

Distributed Databases

Introduction

- “distributed databases” is one form of database organization to represent and process the data. It is a natural extension of client-server and peer-to-peer paradigms.

What is a Distributed Database System?

We define a distributed database as a collection of multiple, logically interrelated databases distributed over a computer network.

A distributed database management system (distributed DBMS) is then defined as the software system that permits the management of the distributed database and makes the distribution transparent to the users.

Distributed Databases

Distributed Database Systems

A distributed database in an environment in which related data sources:

- Are residing on **different geographically distributed sites** — data is closed to the application domain (s) that uses it.
- Might be **replicated** — to improve performance.
- Are **slit** (Fragmented), horizontally and/or vertically, and distributed among the sites — to balance the load and improve performance.

Distributed Databases

Data Distribution

- Data might be distributed to minimize **communication cost** and/or **response time**.
- Data might be kept at the site where it was created to allow **higher control** and **security**.

Distributed Databases

Data Replication

- Data replication allows **more availability** (in the event of failure — if one site is down, data can be accessed from another site),
- Data replication increases the **degree of parallelism** (reduce response time),
- Data replication increases **overhead on update**.

Data replication is the process of storing separate copies of the database at two or more sites. It is a popular fault tolerance technique of distributed databases.

Distributed Databases

Advantages of Data Replication

- **Reliability** – In case of failure of any site, the database system continues to work since a copy is available at another site(s).
- **Reduction in Network Load** – Since local copies of data are available, query processing can be done with reduced network usage, particularly during prime hours. Data updating can be done at non-prime hours.
- **Quicker Response** – Availability of local copies of data ensures quick query processing and consequently quick response time.
- **Simpler Transactions** – Transactions require less number of joins of tables located at different sites and minimal coordination across the network. Thus, they become simpler in nature.

Distributed Databases

Disadvantages of Data Replication

- **Increased Storage Requirements** – Maintaining multiple copies of data is associated with increased storage costs. The storage space required is in multiples of the storage required for a centralized system.
- **Increased Cost and Complexity of Data Updating** – Each time a data item is updated, the update needs to be reflected in all the copies of the data at the different sites. This requires complex synchronization techniques and protocols.
- **Undesirable Application – Database coupling** – If complex update mechanisms are not used, removing data inconsistency requires complex co-ordination at application level. This results in undesirable application – database coupling.

Distributed Databases

Data Replication

- In general, data replication increases the performance of read operations and increases the availability of the data to **read-only transactions**.
- However, data replication incurs overhead for **update transactions**, since all copies of the data must be synchronized. In addition, controlling **concurrent updates** by several transactions on replicated data is becoming very complex.

Distributed Databases

Data Replication

If the distributed database is (partially or fully) replicated, it is necessary to implement protocols that ensure the consistency of the replicas, i.e., copies of the same data item have the same value. These protocols can be *eager* in that they force the updates to be applied to all the replicas before the transaction completes, or they may be *lazy* so that the transaction updates one copy (called the master) from which updates are propagated to the others after the transaction completes.

Distributed Databases

Data Fragmentation

Fragmentation is the task of dividing a table into a set of smaller tables. The subsets of the table are called **fragments**.

Fragmentation can be of three types: **horizontal**, **vertical**, and **hybrid** (combination of horizontal and vertical).

Horizontal fragmentation can further be classified into two techniques: primary horizontal fragmentation and derived horizontal fragmentation.

Distributed Databases

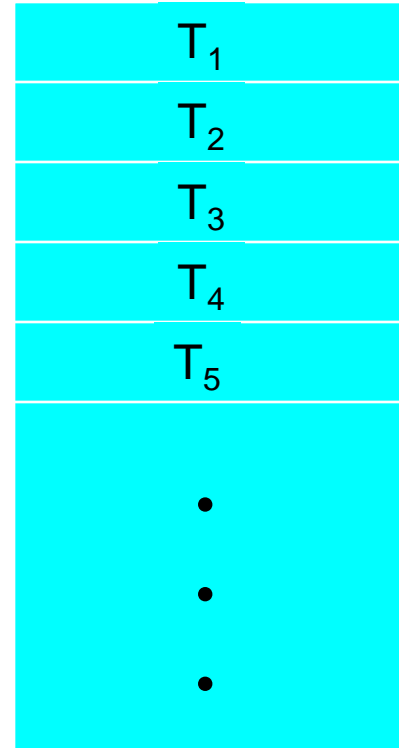
Data Fragmentation — Horizontal fragmentation

- In **horizontal fragmentation**, data set is **partitioned** (broken) into subsets with the same structure — original data set can be constructed by taking the union of the subsets.
- Horizontal fragmentation keeps the subsets at the sites where they are used the most.
- Horizontal fragmentation is **lossless**.

Distributed Databases

Data Fragmentation – **Horizontal fragmentation**

$$T = \cup T_i \quad 1 \leq i \leq n$$



Distributed Databases

Fragmentation should be done in a way so that the original table can be reconstructed from the fragments. This is needed so that the original table can be reconstructed from the fragments whenever required. This requirement is called “reconstructiveness.”

Advantages of Fragmentation

- Since data is stored close to the site of usage, efficiency of the database system is increased.
- Local query optimization techniques are sufficient for most queries since data is locally available.
- Since irrelevant data is not available at the sites, security and privacy of the database system can be maintained.[1]

[1] : https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_design_strategies.htm

Distributed Databases

Disadvantages of Fragmentation

- When data from different fragments are required, the access speeds may be very high.
- In case of recursive fragmentations, the job of reconstruction will need expensive techniques.
- Lack of back-up copies of data in different sites may render the database ineffective in case of failure of a site.

Distributed Databases

Data Fragmentation — Vertical fragmentation

- In **vertical fragmentation**, data set is **decomposed** (broken) into subsets with different attributes — Original data set is constructed by performing join on the fragments.
- Vertical fragmentation is also **lossless**.

Distributed Databases

Data Fragmentation — Vertical fragmentation

$$T_i = \Pi_{L_i}(T) \quad 1 \leq i \leq n$$

$$T = T_1 \bowtie T_2 \dots \bowtie T_n$$

Distributed Databases

Vertical Fragmentation

In vertical fragmentation, the fields or columns of a table are grouped into fragments. In order to maintain reconstructiveness, each fragment should contain the primary key field(s) of the table. Vertical fragmentation can be used to enforce privacy of data.



The End

Thanks for listening ..

Any questions ?