Name : Prof. Mukhtar Ansari
Class : TE
Semester : VI
Course : Distributed Database
Academic year : 2017-18

Slides prepared from : Chhanda Ray , “Distributed Database System”, Pearson Education India.

# Chapter 1 : Concept and Overview Distributed Database system

What is Distributed Database System (DDBS),
Features of DDBS,
promises of DDBS,
Design issue in DDBS,
Distributed DBMS architecture :
Client/server System,
Peer-to-Peer,
Mutli-Database system.

# What is Distributed Database System (DDBS)

## Fundamental of DIstributed Databases :

- Area of Information Processing
- DD is the DB which is under the control of central DBMS. datat stored on multiple computers located ……
- DD must ensure following :
  *Distribution is transparent* : seemless distribution
  *Transactions are transparent* : maintain integrity, transaction can also be divided into sub transactions

# DDBMS

- It consist of single logical db that is split into number of partition or **fragments**.
- each fragment is stored on one or more computers under the **control** of seperate DBMS with all connected by communication network.
- each independent computer will be called as **SITE**.
- each site has **capablity** of preocessing local as well as global reuest via network
- all site work together in such a way that **any** user can access **any** data from **anywhere** via network

DD system allows application to access the data items from local and remote database

Applications are classified into two cat :

1. ***Local Application*** : require data from local site only
2. ***Global Application*** : require data from local as well as remote site

## Features of DDBMS

1. DDBMS is a collection of logically related shared data.
2. data in DDBMS is split into number of fragment.
3. Fragments may be replicated in distributed system.
4. Fragments are allocated to different site.
5. In distributed system sites are linked via network.
6. data in each site is under the control of a DBMS.
7. DBMS at each site has its own rights to handle local application and data.
8. each DBMS in distributed system participate in atleast one global application.

# Advantages of DDBMS

- Sharing of Information : information is shared
- Faster data access : if data is locally
- Speeding up query processing : query split
- Increased local autonomy : responsible for local data
- Increased Availablity : if one site fails ..... data is avail
- Increased Reliablity : if one site fails ..... database is rel
- Better Performance : data stored as per demands
- Reduced operating cost : economy
- Integration of existing db : databses can be clubed
- Processor Independence :
- Modular Extensiblity : new sites can be added

# Disadvantages of DDBMS

- ***Increased Complexity*** : management is difficult {performance, Availablity, Reliablity}
- ***Increased maintenance*** : maintenance cost is higher
- ***Increased commmunication cost*** : required network to communicate to remote sites
- ***Security*** : as data is distributed so probablity of security lapses increases also network must be strong and secure
- ***Lack of Standards*** : all sites may be hetrogeneous in terms of processor, network. it lacks the potential of DBMS

## Disadvantages of DDBMS (Continued .....)

- ***Increased Storage requirement*** : cause of data replication DBMS require additional storage space
- ***Integrity maintenance is very difficult*** :Integrity refers to the VALIDITY and CONSISTENCY of stored data
- ***Lack of Experience*** : compare to the centralized DB people are not much familier and experienced with it.
- ***DB Design more complex*** : disign include fragmentation, allocation and replication so design is more complex.

# Homogeneous and Hetrogeneous Distributed DBMS

**Homogeneous DDBMS :** if all sites uses same DBMS product/software, it is called HDDBMS

**Hetrogeneous DDBMS**: all sites uses different DBMS product/software

- Different Hardware
- Different DBMS software
- Different Hardware and different DBMS software

## Promises of DDBMS

Distributed Database Systems deliver the following advantages:

- Higher reliability
- Improved performance
- Easier system expansion
- Transparency of distributed and replicated data

# Higher reliability

- Replication of components
- No single points of failure
- e.g., a broken communication link or processing element does not bring down the entire
- System Distributed transaction processing guarantees the consistency of the database and concurrency

## Improved performance

- Proximity of data to its points of use
  - Reduces remote access delays
  - Requires some support for fragmentation and replication
- Parallelism in execution
  - Inter query parallelism
  - Intra query parallelism

## Easier system expansion

- Issue is database scaling
- Emergence of microprocessor and workstation technologies
  - Network of workstations much cheaper than a single mainframe computer
- Data communication cost versus telecommunication cost
- Increasing database size

## Transparency

- Refers to the separation of the higher level semantics of the system from the lower level implementation issues
- A transparent system “hides” the implementation details from the users.
- A fully transparent DBMS provides high level support for the development of complex Applications.

# A. Data Independence

Fundamental form of transparency
2 types
**Logical** : the immunity of user applications to changes in the logical structure (i.e., schema) of the database.
**physical** : deals with hiding the details of the storage structure from user applications.

## B. Network Transparency

The user should be protected from the operational details of the network; possibly even hiding the existence of the network.

## C. Replication Transparency

if one of the machines fails, a copy of the data are still available on another machine on the network.

Replication transparency refers only to the existence of replicas, not to their actual location.

## D. Fragmentation Transparency

It is commonly desirable to divide each database relation into smaller fragments and treat each fragment as a separate database.

This is commonly done for reasons of performance, availability, and reliability.

# Complicating Factors

- Complexity
- Cost
- Security
- Integrity control more difficult
- Lack of standards
- Lack of experience
- Database design more complex

# Design Issues in DDBMS

1. Distributed Database Design
2. Distributed Directory Management
3. Distributed Query Processing
4. Distributed Concurrency Control
5. Distributed Deadlock Management
6. Reliability of Distributed DBMS
7. Replication

# Distributed Database Design

How the database and the applications that run against it should be **placed** across the sites.

There are two basic alternatives to placing data : non replicated and replicated.

Replicated designs can be either fully replicated where the entire database is stored at each site, or partially replicated where each partition of the database is stored at more than one site, but not at all the sites.

Non Replicated design the database is divided into a number of partitions each of which is placed at a different site.

The two fundamental design issues are **fragmentation**, the separation of the database into partitions called fragments, and **distribution**, the optimum distribution of fragments.

# Distributed Directory Management

A directory contains **information** (such as descriptions and locations) about data items in the database.

A directory may be **global** to the entire DDBS or **local** to each site.

It can be **centralized** at one site or **distributed** over several sites.

There can be a **single** copy or **multiple** copies.

# Distributed Query Processing

Query processing deals with **designing** algorithms that **analyze** queries and **convert** them into a series of data manipulation operations.

The problem is how to decide on a **strategy** for executing each query over the network in the most **cost** effective way, however cost is defined.

The **factors** to be considered are the distribution of data, communication costs, and lack of sufficient locally available information.

# Distributed Concurrency Control

Concurrency control involves the **synchronization** of accesses to the distributed database, such that the **integrity** of the database is maintained.

In Distributed enviroment not only has to worry about the integrity of a single database, but also about the consistency of multiple copies of the database.

The alternative solutions are 2 general classes :

**Pessimistic** , synchronizing the execution of user requests before the execution starts, and

**Optimistic**, executing the requests and then checking if the execution has compromised the consistency of the database.

Two fundamental primitives that can be used with both approaches are **locking**, which is based on the mutual exclusion of accesses to data items, and **timestamping**, where the transaction executions are ordered based on timestamps.

# Distributed Deadlock Management

The deadlock problem in DDBSs is similar in nature to that encountered in operating Systems.

The competition among users for access to a set of resources (data, in this case) can result in a deadlock if The Synchronization mechanism is based on locking.

The well known alternatives of prevention, avoidance, and detection/recovery also apply to DDBSs.

# Reliability of Distributed DBMS

We mentioned earlier that one of the potential advantages of distributed systems is improved reliability and availability.

It is important that mechanisms be provided to ensure the consistency of the database as well as to detect failures and recover from them.

The implication for DDBSs is that when a failure occurs and various sites become either inoperable or inaccessible, the databases at the operational sites remain consistent and up to date.

when the computer system or network recovers from the failure, the DDBSs should be able to recover and bring the databases at the failed sites up to date.

This may be especially difficult in the case of network partitioning, where the sites are divided into two or more groups with no communication among them.

## 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 DBMS Architecture :

The architecture of a system defines its structure.

This means that the components of the system are identified, the function of each component is specified, and the interrelationships and interactions among these components are defined.

1. Client/Server Systems
2. Peer to Peer Systems
3. Multidatabase System Architecture

# Client/Server Systems

The Client server system is versatile, message based and modular infrastructure that is intended to improve usablity, flexiblity, interoperablity and scalablity as compared to centralized, mainframe, time sharing computing.

## Client

It is an individual user's computer or user application that does a certain amount of processing on its own and sends and receives requests to and from one or more servers.

## Server

It consist of one or more computers or an application program that receives and processes requests from one or more client machines

## Advantages of Client/Server System

Client is the requester of services

Advantages :

1. ability to distribute the workload
2. allow end user to use GUI hence improve functionality and simplicity
3. better performance

# Disadvantages of Client/Server System

Server is provider of services

Disadvantages :

1. more complex enviroment (OS Platform)
2. clients are distributed over many places
3. may suffer from security problem (clients increases)
4. maintenance cost is more

# Funtions of DDBMS

1. Application Interfaces
2. Validation and transformation techniques
3. Distribution Transparency
4. Mapping and I/O Interface
5. Management of replicated data
6. Extended communication services
7. Extended query processing and optimization
8. Distributed Transaction Management
9. Distributed backup and recovery services
10. Distributed COncurrency Control
11. Support for global system catalog
12. Support for global Database Administrator
13. Distributed Security Control

# THANK YOU