MIS 301 RELATIONAL DATABASE MANAGEMENT SYSTEM

DATABASE MANAGEMENT SYSTEM

Advanced Data Models & Emerging Trends : Advanced Data Modeling Concepts, Object–Oriented Databases, Distributed Databases & Client Server Architecture, XML

Lecture 32&33

OBJECT-ORIENTED DATABASE

- Object oriented database management systems are based on objects in object-oriented programming
- In object-oriented programming, an entity is represented as an object and objects are stored in memory.
- Objects have members such as fields, properties, and methods
- Objects have a life cycle that includes the creation of an object, use of an object, and deletion of an object
- Encapsulation, inheritance, and polymorphism are properties of object oriented programming

FEATURES OF OBJECT-ORIENTED DATABASE

- Object oriented databases provide non-volatile storage to objects
- Object databases are difficult to conceptualize for non-programmers
- Unlike relational databases, object-oriented databases do not have a standard language, like SQL, associated with them
- They combine database capabilities with object-oriented programming language capabilities
- Object Database programmers can replicate or modify existing objects to make new objects reducing their work
- Access to data can be faster because an object can be retrieved directly without a search, by following pointers
- An object is a concrete unit in an abstract class

Advantages & Disadvantages of Object-Oriented Databases

Advantages	Disadvantages
Complex data sets can be saved and retrieved quickly and easily.	Object databases are not widely adopted.
Object IDs are assigned automatically	In some situations, the high complexity can cause performance problems.
Works well with object-oriented programming languages.	They are comparatively difficult to pick up

OODB

- The combination of relational model features (concurrency, transaction, and recovery) with object-oriented principles(Encapsulation, Polymorphism, Inheritance) results in an object-oriented database model.
- **Object-Oriented Programming + Relational Database Features = Object-Oriented Database Model**

FEATURES OF OBJECT ORIENTED DATABASES

- **Object** :- An **object** has properties (state) and methods (behaviour). Each object is identified using a unique object identifier.
- **Encapsulation**:-The implementation details are hidden from the end-users and only the needed descriptions are shown.
- **Inheritance**:- It enables re-usability by creating new classes from existing classes. The derived classes inherit the properties of their parent class and also have their own unique properties

FEATURES OF OBJECT ORIENTED DATABASES

- **Association:** Association refers to links between the various entities of an application.
- **Polymorphism** *:* It allows several objects to represent the same message in different ways.

STRENGTHS OF OBJECT ORIENTED DATABASES

- OODBs allow for the storage of complex data structures that cannot be easily stored using conventional database terminology.
- OODBs support all the features required for object oriented applications.
- OODBs contain active object servers which support both distribution of data and distribution of work.

STRENGTHS OF OBJECT ORIENTED DATABASES

- OODBs support reusability. Generic objects can be defined and then reused in numerous application.
- OODBs can manage complex data types such as document, graphics, images, voice messages, etc.
- Due to the existing mode of communication between objects, OODBMS can support distribution of data across networks more easily thus supporting distributed databases.

ADVANTAGES OF OODB OVER RDBMS

- Objects do not require assembly and disassembly saving coding time and execution time to assemble or disassemble objects.
- Reduced paging
- Easier navigation
- Better concurrency control
- Data model is based on the real world
- Works well for distributed architectures
- Less code required when applications are object oriented.

DISTRIBUTED DATABASES

- In a distributed database, there are a number of databases that may be geographically distributed.
- A distributed DBMS manages the distributed database in a manner so that it appears as one single database to users.
- In summary, a **distributed database** is a collection of multiple interconnected databases, which are spread physically across various locations that communicate via a computer network.
- Databases in the collection are logically interrelated with each other.
- A **distributed database management** system (DDBMS) is a centralized software system that manages a distributed database in a manner as if it were all stored in a single location.

ADVANTAGES OF DISTRIBUTED DATABASES

- **Modular Development** – If the system needs to be expanded to new locations or new units, the work simply requires adding new computers and local data to the new site and finally connecting them to the distributed system, with no interruption in current functions, unlike centralized systems.
- **More Reliable** – Unlike centralized systems, in distributed systems, when a component fails, the functioning of the system continues , sometimes, at a reduced performance. Hence DDBMS is more reliable.
- **Better Response** – If data is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response.
- **Lower Communication Cost** – In distributed database systems, if data is located locally where it is mostly used, then the communication costs for data manipulation can be minimized.

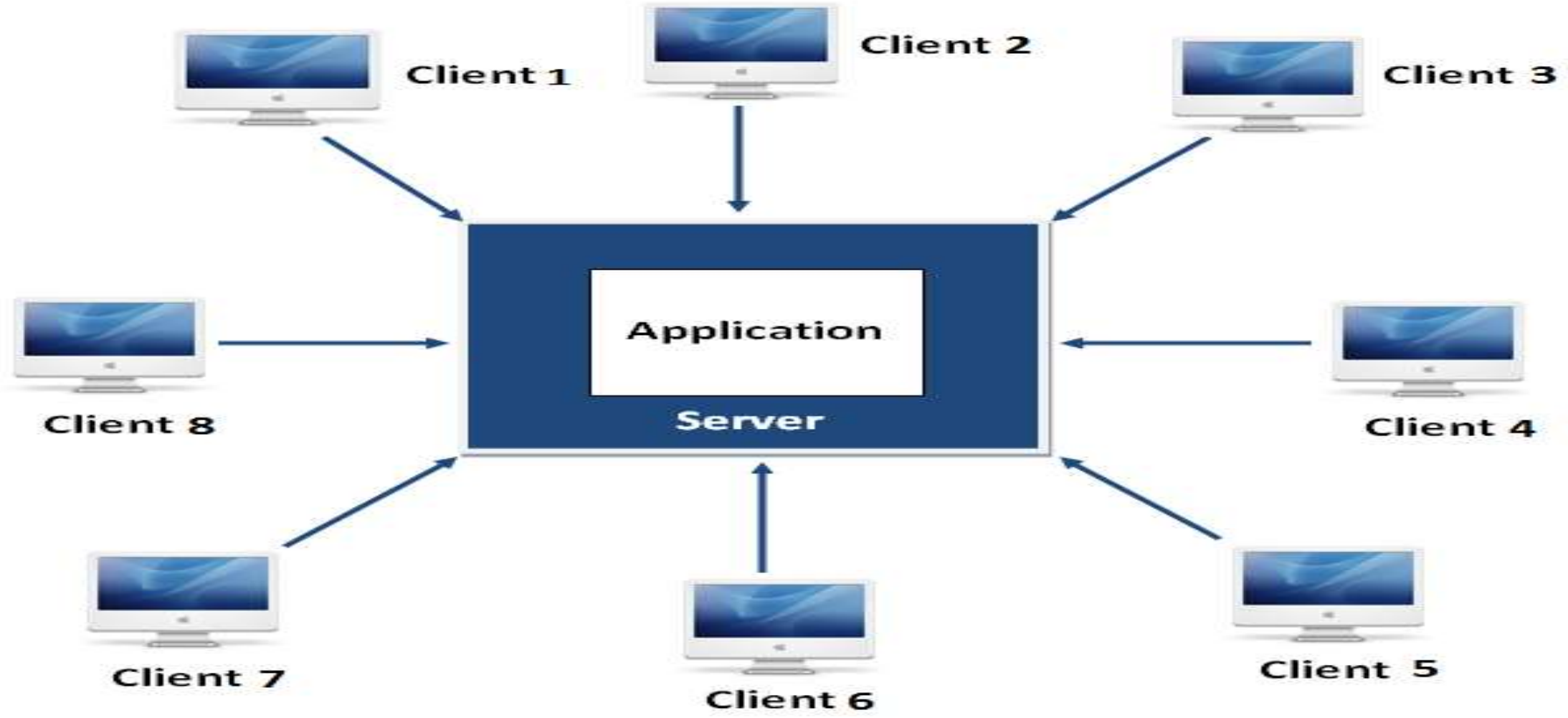
DISADVANTAGES OF DISTRIBUTED DATABASES

- **Need for complex and expensive software** – DDBMS demands complex and expensive software to provide data transparency and co-ordination across several sites.
- **Processing overhead** – Even simple operations may require a large number of communications and additional calculations to provide uniformity in data across the sites.
- **Data integrity** – The need for updating data in multiple sites pose problems of data integrity.
- **Overheads for improper data distribution** – Responsiveness of queries is largely dependent upon proper data distribution. Improper data distribution often leads to very slow response to user requests.

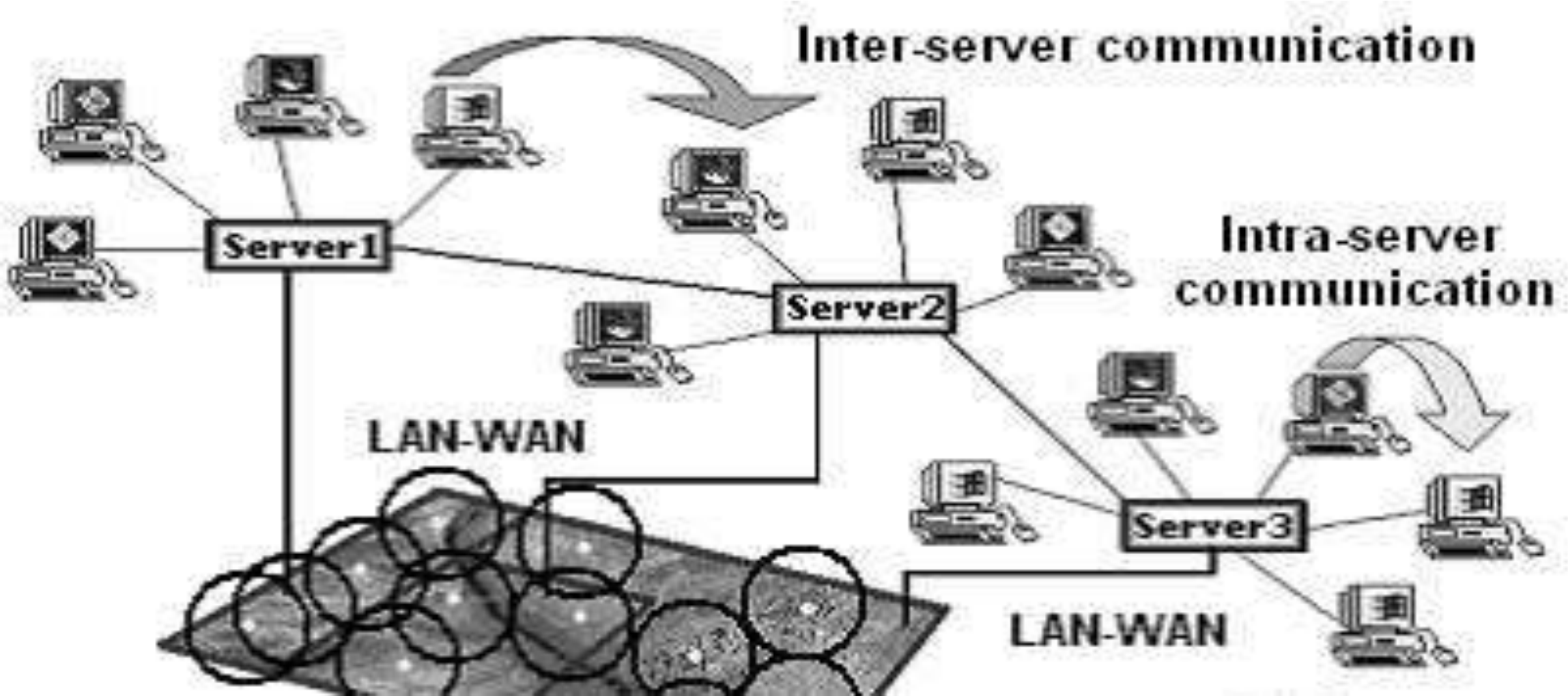
CLIENT - SERVER ARCHITECTURE FOR DDBMS

- This is a two-level architecture where the functionality is divided into servers and clients.
- The server functions primarily encompass data management, query processing, optimization and transaction management.
- Client functions include user interface, consistency checking and transaction management.
- The two different client - server architecture are –
  i. Single Server Multiple Client
  ii. Multiple Server Multiple Client

SINGLE SERVER MULTIPLE CLIENT



MULTIPLE SERVER MULTIPLE CLIENT



XML

- Extensible Markup Language (**XML**) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable
- XML stores and transports data
- XML is designed to be self-descriptive. It contains sender information, receiver information, a heading and a message body.
- XML was designed to carry data - with focus on what data is while HTML was designed to display data - with focus on how data looks
- The XML language has no predefined tags unlike HTML. With XML, the author must define both the tags and the document structure.
- XML is Extensible. Most XML applications will work as expected even if new data is added (or removed).
- It simplifies data sharing, data transport, platform changes and data availability.
- XML stores data in plain text format. This provides a software and hardware independent way of storing, transporting, and sharing data.

- TILL WE MEET AGAIN IN THE NEXT CLASS.....