Advanced Databases.

Module 1- Parallel and Distributed Databases.

* Parallel Databases.

→ Parallel Database is a DBMS that runs through multiple processors and disks. They combine two or more processor also disk storage which allows operations and executions to occur faster.

→ They are mainly designed for concurrent operations.

* Architecture of Parallel DBMS.

① Shared Memory System.

A shared memory system is an architecture of DBMS where a computer processor is able to process data from multiple memory modules or unit through intercommunication channel.

Advantage of S.M

a) Data can be easily accessed from various processors.

b) A single processor can send messages to other processors efficiently.

→ Disadvantages of S.M.

a) Waiting time for every single processor increases when all are in use.

b) Band width is an issue.

(2) Shared Disk system.

Shared disk system allows computer processors to access multiple disks through intercomm. network. can also access and utilize every local memory.

→ Advantages.

a) Fault tolerance can be achieved.

→ Disadvantages.

a) Addition of processors slow down existing ones
b) Limited scalability.

(3) Shared Nothing System.

When all the processors have their own disks and memory for the objective of efficient work flow.

Advantages

a) This system has more scalability.
b) No. of processors and disks are connected as per requirement.

Disadvantages

a) Requires partioning of data.
b) Cost of every system is high.

* Advantages of Parallel DBMS

① Speed - The servers from parallel DBMS are able to break up user database request into parts and dispatch each parts to seperate computers. They work on it seperately and merge the result together.

This allows faster access to large databases.

② Reliability.

A parallel database when properly configured, can continue to work despite the failure of any computer in the cluster.

Database server can sense that a specific computer is not responding, ner it can reroute the work to other computers.

③ Capacity.

The third advantage is capacity. As more and more users request access to the database, the computer admin adds more computers to the parallel server.

This boosts the overall max capacity and then thousands of requests can be handled concurrently.

* Disadvantages of Parallel DBMS.

① Cost.

For quicker response and efficient searching, more and more disks and processors need to work simultaneously, so its never cheap to implement parallel DBMS

② Resources

A lot of disks, processors and memory is required to manage large databases. This also takes into consideration that replacement of these resources is also necessary.

(5) Difficulty in Managing Systems:

There are a lot of systems running at the same time. This becomes hectic and may lead to misbalance of resources.

---

* Query Evaluation in Parallel DBMS.

There are mainly 5 techniques of Query Evaluation

1. I/O Parallelism
2. Intra-query parallelism
3. Inter-query parallelism.
4. Intra-operation parallelism
5. Inter-operation parallelism.

Parallelism allows parallel execution of multiple queries by decomposing them into parts that work in parallel.

Mostly achieved in shared-Nothing architecture.

1) I/O parallelism.

→ Input/output parallelism is ~~He~~ a key technique used in parallel database management.

→ This involves breaking up the I/O operations into smaller independent tasks that can be executed in parallel across multiple nodes. This is done mainly by data partitioning.

→ There are 4 types of partioning in I/O parallelism:

a) Hash partitioning.

Hash is a fast mathematical function. Each row has an index which can be hashed to get towards desired data faster.

b) Range partitioning

Suppose there are 100 values and 4 disks, disk 0, disk 1, disk 2, disk 3.

0 - 25 values → DISK 0
26 - 50 values → DISK 1
51 - 75 values → DISK 2
76 - 100 values → DISK 3

c) Round - Robin partioning

In this type of partitioning relations are studied in order, every ith type is sent to disk no. (i%n). So disk turns recieve new data.

d) Schema - Partitioning

Here different tables in the database are placed on different disks.

---

② Intra - Query Parallelism.

→ Unlike I/o parallelism, Intra Query breaks down the query instead of operations to smaller tasks like sorting, join etc. and are executed in parallel.

→ This has two approaches.

a) First → Each CPU can execute the duplicate task against some data portion.

b) Second → Tasks can be divided into different sectors with each CPU executing a distinct sub task.

③ Inter Query Parallelism.

→ Here, execution of multiple transactions like in the CPU is made faster and more efficient.

→ It involves dividing the available processing resources among multiple queries, allowing them to be executed in parallel.

→ Now the DBMS allocates different processors with different queries making it easier to implement.

---

④ Intra-operation Parallelism.

→ Here each individual operation of a task like sorting, join, projection etc. is parallelized.

⑤ Inter-operation Parallelism.

→ This involves assigning multiple operations to multiple processors to operate at parallel.

* Parallelizing individual operations

This process includes several steps:-

① Task decomposition - Involves breaking down the operation into smaller, independant tasks that are to be executed in parallel.

② Resource Allocation - Involves assigning the tasks to specific processors and nodes based on availability and capacity.

③ Task scheduling :- Involves determining the order in which the tasks should be executed, taking into account their dependancies and resource requirements.

④ Result Merging :- It involves combining of the results of the parallel tasks into a single output and returned to the user.

* Distributed Database.

A distributed database is a collection of multiple interconnected databases which are spread physically across various location that communicate via a computer network.

* Features of Distributed DBMS.

1) Databases in the collection are logically inter-related with each other. often they represent a single logical database.

2) Data is physically stored at multiple sites.

3) Makes sure that data modified at any point is universally updated.

4) Maintains confidentiality and data integrity

* Why chose Distributed DBMS.

① Distributed Nature - Most organisations currently are sub-divided into multiple units and each unit has its own set of data.

② Need for sharing data - Multiple organisation often need to communicate with each other and share data and resources.

③ Support for OLAP and OLTP - Online Transaction Processing (OLTP) and Online Analytical Processing (OLAP). DDBMS supports these processes by providing synchronised data.

④ Database Recovery - DBBMS has data replicated at multiple sites, means that losing data at a site will not be fatal. Users can access data at different site while the damaged one is being repaired.

⑤ Support for Multiple Application Software :- Most organisation use a variety of application softwares, but DDBMS provides a uniform functionality for using the same data among different platforms.

---

* Advantages of DDBMS.

① More Reliable - In case of database failures, the total system of centralised databases comes to a halt, but in DDBMS data is stored at various places so data is obtained but may be at reduced speeds.

(2) Modular development - If a system needs to be expanded to new locations or new units. In centralized database systems, the action requires substantial efforts and disruptions, but in DDBMS, the work is simply adding new hosts and connecting them to the distributed system.

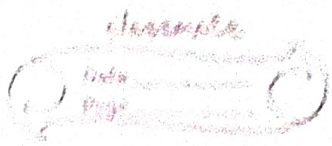
(3) Better Response - In DDBMS, the query is processed in the local data itself, providing faster response, However, in centralised systems, the query is processed at the central computer, taking longer time.

---

* Disadvantages of DDBMS

(1) Need for complex and expensive software :- DDBMS requires complex and expensive software to provide data transparency and coordination across several sites.

(2) Processing overhead - Even simple operations may require large number of communication and calculation.



③ Data Integrity - The need for updating data in multiple files pose problems in data integrity

---

* Types of DDBMS

① Homogeneous DDBMS

In homogeneous DDBMS, all nodes in the network run the same DBE and also use the same model. This means that each node is capable of processing the same type of data and executing the same set of operations.

Here the data is divided into fragments and distributed across multiple nodes. Nodes are connected to the network.

→ Advantages -

① Easier to manage as every database uses same model and database system.

② Often used where there is a need of high availability and fault tolerance.

Disadvantages

① Not flexible - Different softwares which requires different systems might not integrate with homogeneous systems.

② High cost , - Need high maintainence and high degree of coordination to implement at full potential.

---

② Heterogeneous DDBMS.

→ In a heterogeneous DDBMS, different nodes in the network have different database systems and use different data models.

→ These are often used when multiple third-party applications need to be integrated within the DDBMS.

→ Each node is capable of processing different types of data and executing different sets of operations.

→ Advantages

a) More flexible - More supportive towards third-party applications along with integration.

b) Performance - Much better performance for specific data types and applications.

→ Disadvantages

a) Complex systems, - As the data models and systems are different for each node, coordination is much more complex and costly.

b) Data Inconsistency - Since data is being processed in a non-uniform way, inconsistency in data is a viable issue.

---

* Distributed DBMS Architecture.

Architectures are mainly developed based on 3 factors

① Distribution
② Autonomy
③ Heterogeneity

* Architecture Models.

→ Some common architecture models are:-

① Client - Server Architecture

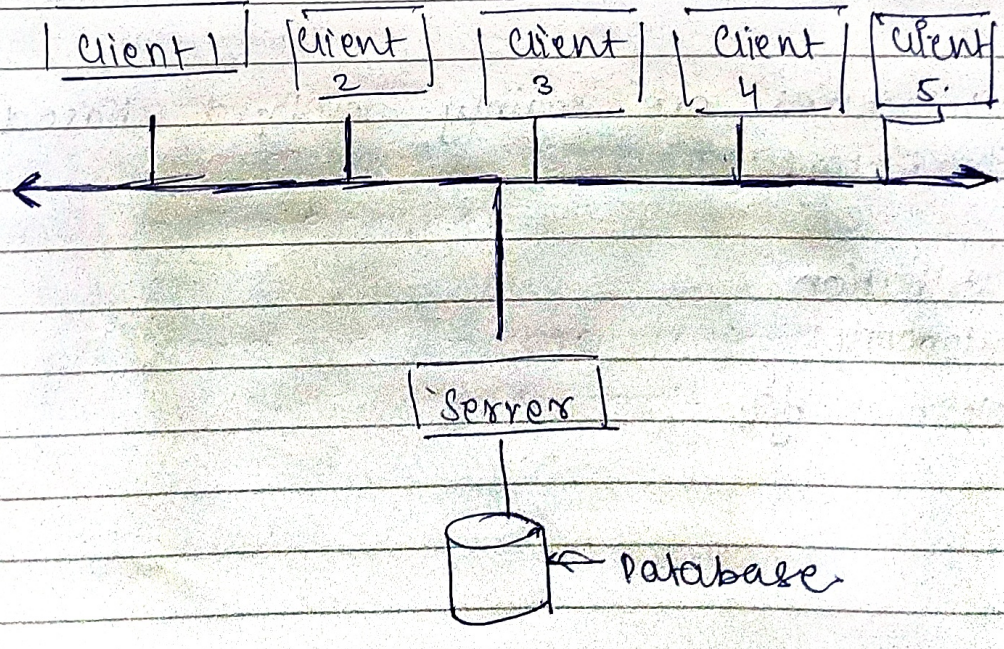
This is a two-level architecture where the functionality is divided into servers and clients.

Server functions include query processing and optimisation, Transaction Management, data management.

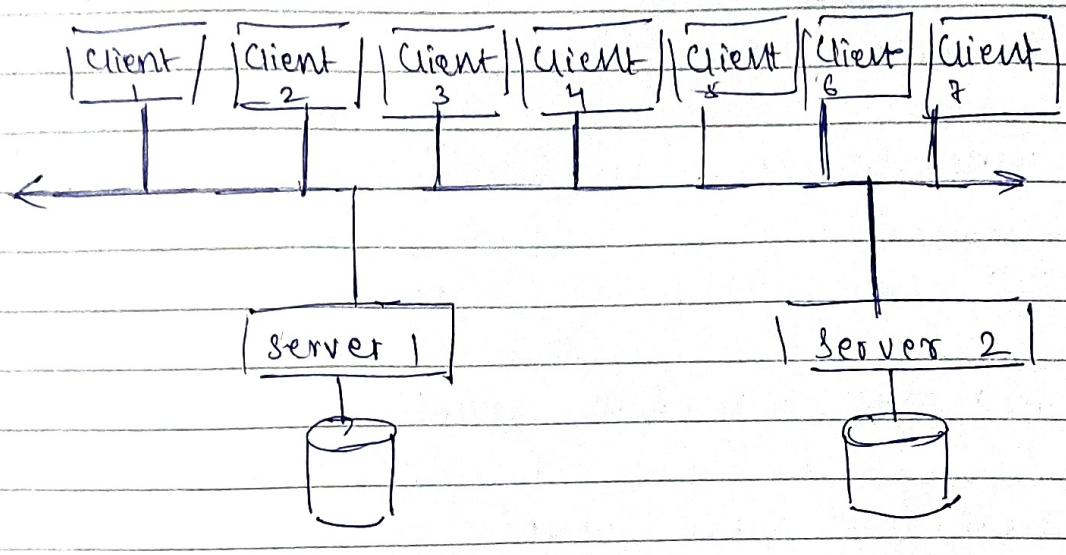
Client functions include, user interface,

These architectures have two types

a) Single Server - Multi Client



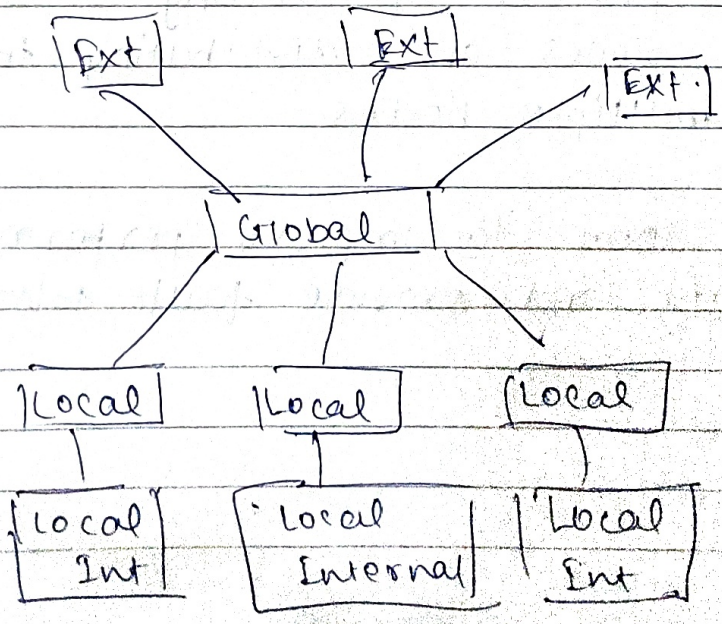
b) Multi Server - Multi client



② Peer-to-peer Architecture

Has 4 levels of schema.

1) Global view
2) Local view
3) External view
4) Local Internal view.



③ Multi-DBMS Architecture

→ A collection of two or more autonomous systems

① Multi-DB View Level

② Multi-DB Conceptual Level

③ Multi-DB Internal Level.

④ Local DB View Level

⑤ Local DB Conceptual Level

⑥ Local DB Internal Level.

* Data fragmentation in DDBMS.

→ Data fragmentation in distributed databases is the process of dividing a database into smaller pieces and distributing these pieces across multiple nodes.

→ This is done to improve performance, increase scalability and provide fault tolerance.

Types of Data Fragmentation.

There are 3 types of data fragmentation.

(1) Vertical Fragmentation:

This technique involves dividing a table into subset of columns and each column is stored in a seperate nodes. This technique is useful only when the queries involve a subset of table column's.

(2) Horizontal Fragmentation.

This technique involves dividing a table into subsets of rows, where each subset is stored on a seperate node. This technique is useful only when the queries involved is a subset of table columns.

(3) Hybrid / Mixed Fragmentation.

This technique is a combination of both horizontal and vertical fragmentation where the table is divided into subset of both rows and columns.

→ Advantages of Data Fragmentation.

① Efficiency of system increases.

② Maintains security and privacy.

→ Disadvantages of Data Fragmentation.

① Recursive Fragmentation can be expensive

② Data from different fragments may be needed.

---

* Data Replication in DDBMS.

→ Data Replication in DDBMS is the process of creating multiple copies of data and storing them on multiple nodes in a distributed system

Some techniques of data replication are :-

① Eager Replication - This involves replicating data immidiately after it is updated on the primary node. Ensures all the data stays up-to-date but this can incur high overhead.

(3) Lazy Replication - This method delays replication of data till it has been requested by the client. This reduces overhead but may store outdated data.

(4) Hybrid Replication - This is a combination of Eager and Lazy Replication where some data is regularly updated while some data is lazily updated.

→ Advantages of Data Replication.

(1) Increases Availability as data is available at multiple nodes.

(2) Provides fault tolerance.

→ Disadvantages of Data Replication

(1) All replicas have to be updated, creating inconsistency

(2) Additional storage requirements.

* Data Allocation in DDBMS.

→ Data Allocation in DDBMS is a process of determining how data is distributed and stored at multiple nodes in the system.

Some techniques in Data Allocation are:-

① Centralized Allocation :- Involves having a single node managing the distribution of data. This creates a single point of failure but simplifies management.

② Decentralized Allocation - Allows each node to manage its own data allocation. This provides fault tolerance but leads to unbalanced load distribution.

③ Hybrid Allocation - A combination of centralised and decentralised allocation. The central node manages overall allocation and each node manages its own allocation.

→ Advantages of Data Allocation.

① Improved performance by minimizing the amount of data that transfers between nodes.

④ Reduced Network Traffic by

→ Disadvantages of Data Allocation

① Increases complexity

② Higher overhead.

---

* Query Processing and Optimization in DDBMS.

① Query Processing

This technique involves coordinating the processes of multiple nodes. The goal of query processing is to ensure efficient and low-cost communication in distributed DBMS.

Query Processing Involves several steps:-

① Query Parsing - This step involves the decomposition of the query into smaller sub-queries that can be executed independently.

② Query optimization - In this step, the query optimizer evaluates, different query execution plans and selects the most optimal plan.

③ Query Execution - Here all the independent sub-queries are run in parallel at different sites in the database.

④ Result Merging - In this step, all the evaluated results are finally merged and sent back to the client.

* Data Transfer Costs.

It is mostly calculated in terms of size of messages.

Using formula DTC = C * size

C = cost per byte
S = size of no. of bytes.

* Query Optimization

Reducing the transfer cost leads to query optimization. Several techniques are used here :-

① Data Partitioning
② Indexing
③ Replication
④ Load balancing

* Concurrency control and Recovery in DDBMS.

① Distributed Two-Phase Locking

→ This is the same as the normal lock protocol seen in regular DBMS, however, in DDBMS each site is assigned as a lock manager, controlling lock acquisition requests from transaction managers.

→ To overlook all the sites, one site is given the authority to look over all the sites and see all transactions.

It can be of three approaches.

① Centralized Two Phase Locking - one site is designated as central lock manager. All the sites in the environment know the location of the CLM and obtain locks from it.

② Primary copy Two-Phase Locking - Many sites are designate as lock managers. Each site has a defined set of locks to manage

③ Distributed Two-Phase Locking - There are many lock managers, each lock manager controls locks of data items stored at local site.

② Distributed Timestamp Concurrency.

→ In distributed systems, any site's global timestamps cannot be from physical/logical clock readings.

→ So it is mostly a combination of the site ID and site's clock reading.

→ Each site has a scheduler that maintains a seperate queue for each transaction manager. During transaction, Lock req. is sent to the site scheduler.

→ Then the scheduler puts the req. in the corresponding queue in increasing timestamp order, oldest first.

---

③ Optimistic Concurrency Control in DDBMS.

There are two rules which extends the optimistic concurrency in regular DBMS.

① Transaction must be validated locally at all sites when it executes, if the transaction is found invalid at any site, it is aborted.

(2) After the transaction passes the first rule, it should be globally validated. Global validation ensures that if two conflicting transactions run together at more than one site, they should commit in the same relative order at all sites.

---

* Recovery in DDBMS:

→ Some common types of failures and recoveries

(1) Recovery from Power Failure

→ Power failure causes loss of information in the non-persistent memory. On power return, all operations are restarted.

→ Transactions on the active list and failed list are undone and written in abort list

→ Transactions in before-commit list are redone.

(2) Recovery from Disk Failure

→ A disk failure or hard crash causes a total database loss. To recover from this a new disk is prepared and all systems are restarted.

→ Transactions in commit list and before commit list are redone.

→ Transactions in active and failed list are written onto the abort list.

③ Checkpointing.

Checkpoint is a point of time at which a record is written onto the database

There are two types of checkpointing

① Consistent Checkpointing

Creates a consistent image of the database at checkpoint. During recovery, transactions on the right side of the last checkpoint are undone and redone. On the left side there are already committed transactions and are not processed again.

② Fuzzy checkpointing

At the time of checkpoint, all the transactions are written in a log, so incase of power failure, recovery manager processes only the transactions which were active during checkpoint.

* Recovery using UNDO/REDO.

This is done to remove all faulty transactions rather than recover from a failure.

This is a two step process.

a) Undo all faulty transactions and transactions that may be affected by the faulty transactions

b) Redo all transactions which were not faulty but have been undone.

---

* In-Memory Databases.

In-memory database is a type of DBMS where all the data is stored in the primary memory (RAM) so all data is lost at power failure

The main advantage of In-memory DB is speed. Data can be accessed super fast as the data is within RAM rather than the disk

These are also designed to support high concurrency and multi-threaded access. Supports all of the ACID properties

One of the main disadvantages of In-Memory DBMS is the scalability, there is a limited amount of data which can be stored in RAM.

Also all data is lost at power failure leading to constant need of recovery management.

---

* Architecture of In-Memory DBMS

① Memory Management - This DBMS uses specialised algorithms to allocate and de-allocate memory when needed.

② Data Storage - Uses B-trees, hashes, and columnar storage to optimize the storage and retrieve of data in memory.

③ Data Access - Supports high speed data access methods like direct memory access or pointer-based access along with caching techniques like page caching or block caching.

(4) Query Optimization - Evaluation of the best query plan to execute queries within minimum storage requirement.

(5) Transaction and concurrency - Utilization of lock-based methods to create a sharing environment.

In-Memory DBMS	DISK DBMS
(1) Data is stored entirely in (RAM).	(1) Data is stored on disk or any other secondary storage.
(2) Access time is much faster	(2) Access time is slower.
(3) Designed to support high concurrency and multi-threaded access	(3) May not be optimized for high concurrency access.
(4) More expensive due to high cost of RAM	(4) Cost effective due to low cost of disk storage
(5) Limited scalability	(5) Can support large datasets so high scalability.