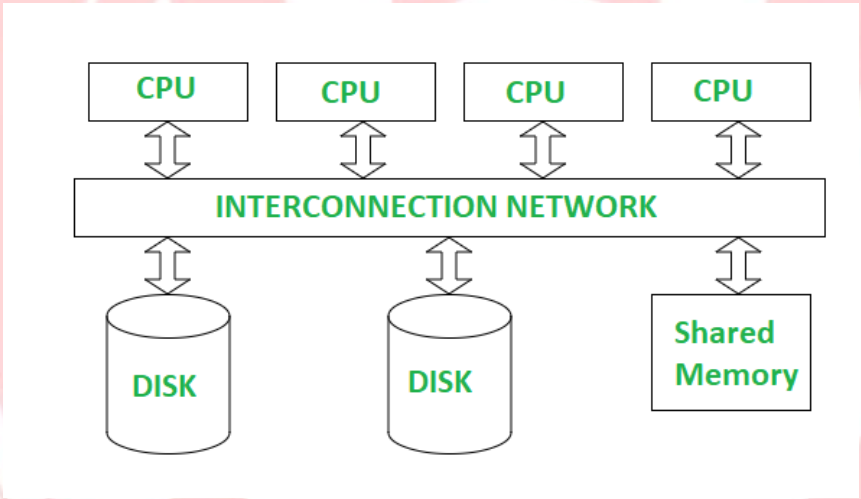
Unit II - Parallel & Distributed Database System

Q) What is parallel database system? Explain its advantages and disadvantages.

Parallel System

"architecture that allows multiple processors or computing units to perform tasks simultaneously in order to speed-up the overall computation process"

- This system can handle large volume of data, speedup queries and improved performance
- Large tasks are divided into small tasks
- Small tasks are executed concurrently on several CPUs



Advantages:

- **High availability:** data is available at parallel DB systems- in case of failure
- **Better performance:** speed up and scale up as number of processors available
- **High processing speed:** as queries are broken down into subtasks and handled by different processors
- **Reliability:** if one computer fails other will be rerouted to finish work
- **Greater flexibility:**
- **Serves multiple users:** single system serves thousands of users

Disadvantages:

- Startup cost high
- If existing CPU gets slower more CPU are added
- Number of resources required are increased
- Complexity is increased

Q) Types of parallel database architectures:

1. Shared memory architecture
2. Shared disk architecture
3. Shared nothing architecture
4. Hierarchical architecture

Shared memory architecture:

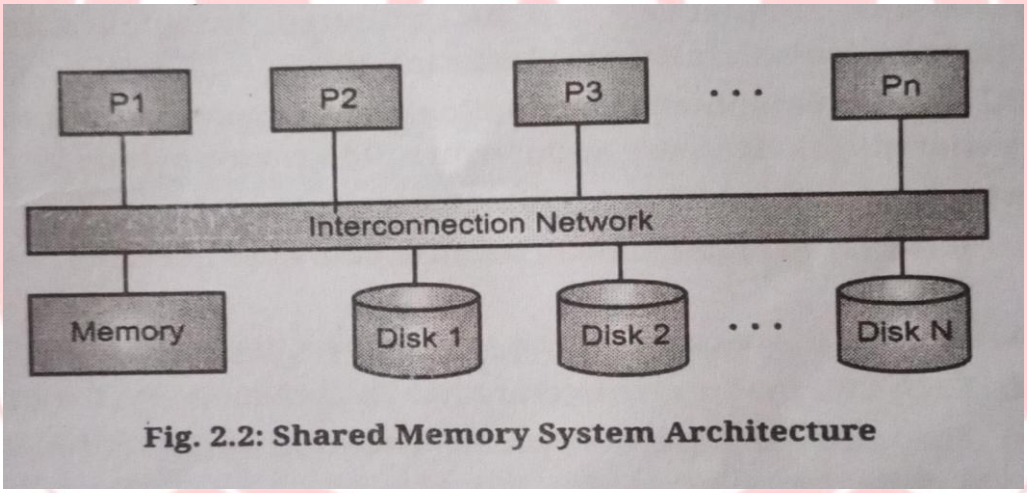


Fig. 2.2: Shared Memory System Architecture

- Multiple CPU attached to interconnection network
- Single / global main memory
- Common disk array (Storage)
- Also known as symmetric multi-processing (SMP)
- Processors can send messages using memory rights in microseconds
- **Not scalable beyond 64 processors**
- Ex. XPRS, DBS3 Volcano

Advantages

- Simple and provides load sharing
- Communication overhead is low
- Communication between CPUs is extremely efficient
- Provides high speed data access

Disadvantes

- Limited extensibility & low availability
- High cost
- Interconnection networks become bottleneck as number of CPUs increases
- Addition of CPU increases waiting time for their turn on bus to access memory.

Shared disk architecture

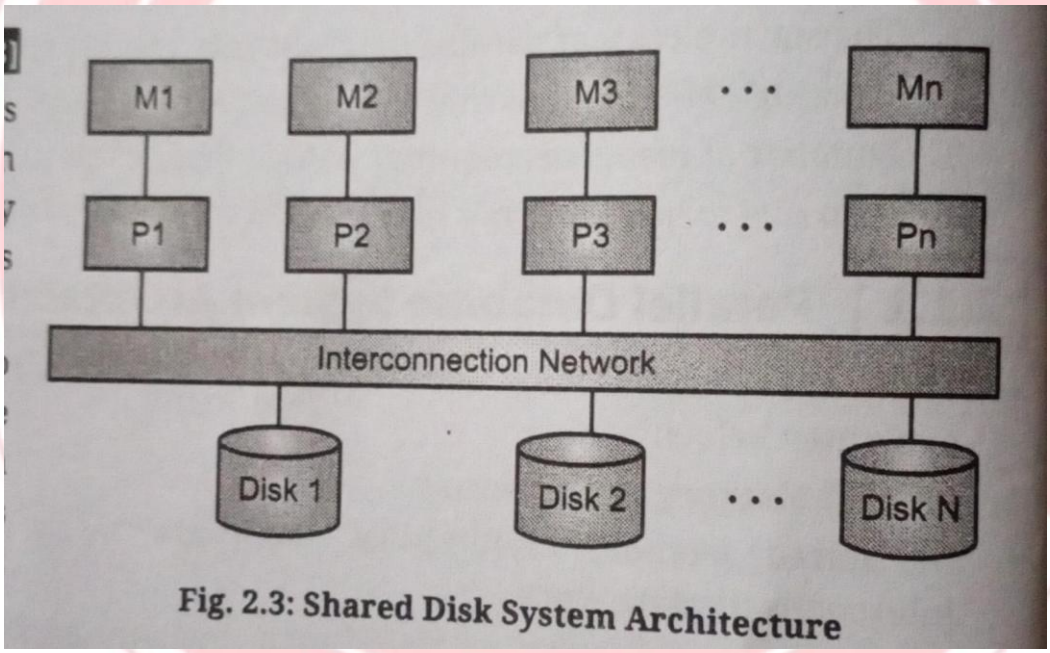


Fig. 2.3: Shared Disk System Architecture

- Multiple CPUs with own memory
- All have access to same disk
- Memory is not shared among CPUs
- Inherently centralized application
- Each CPU can assess all disks but have own private memory
- Also named as **CLUSTERS.**
- Commercial Users **DEC** (Digital Equipment Corporation) – Cluster running Rdb

- Rdb is owned by Oracle – called Oracle Rdb

Advantages

- Easy to load balance
- Low-cost solution to provide degree of fault tolerance
- Each CPU has own memory so bus is not a bottleneck
- Less cost than shared memory architecture
- High performance, high availability, high extensibility.

Disadvantages

- More complex as it requires distributed database
- Communication across processors is slower
- Scalability can lead to bottleneck

Shared nothing architecture

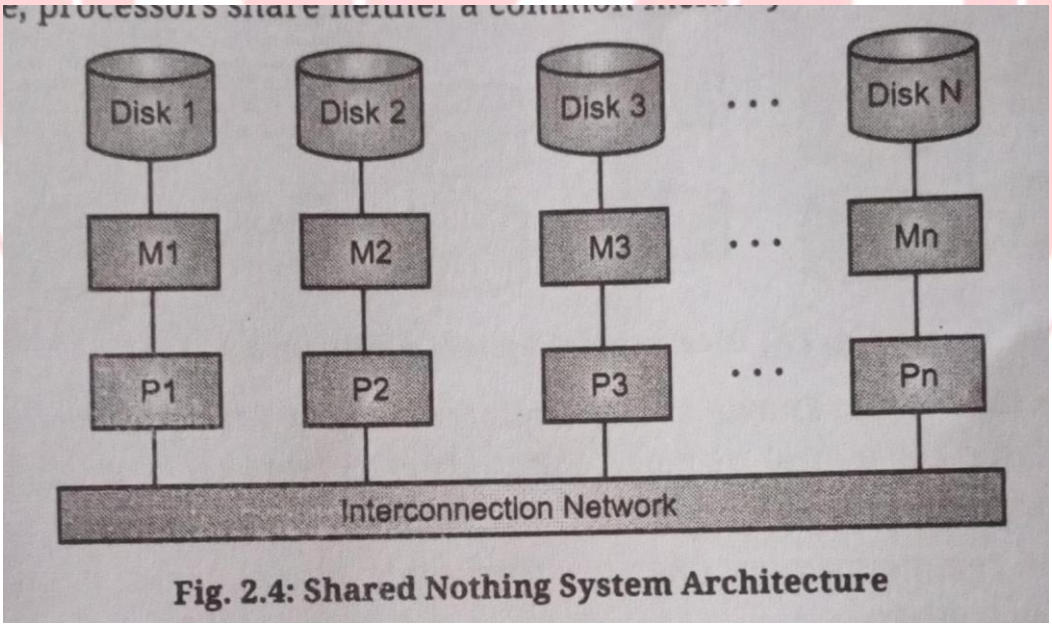


Fig. 2.4: Shared Nothing System Architecture

- Processors have own memory and disk
- NO sharing of memory and disk
- Each CPU has own copy of OS and DBMS

- Also called as massively parallel processing (MPP)
- Communication using high speed interconnection network
- Commercial user – Teradata database machine, Grace, Gama research prototype

Advantages

- Highly scalable
- Provides linear speed – up and linear scale-up
- Easily supports large number of CPUs
- Extensible
- Repication of data on multiple nodes proves high scalability

Disadvantages

- Difficult to load balance
- Requires OS capable of handling heavy amount of messaging to support inter processor
- Communication
- Higher cost of communication with non-local disk
- More complex – for large number of nodes

Hierarchical architecture

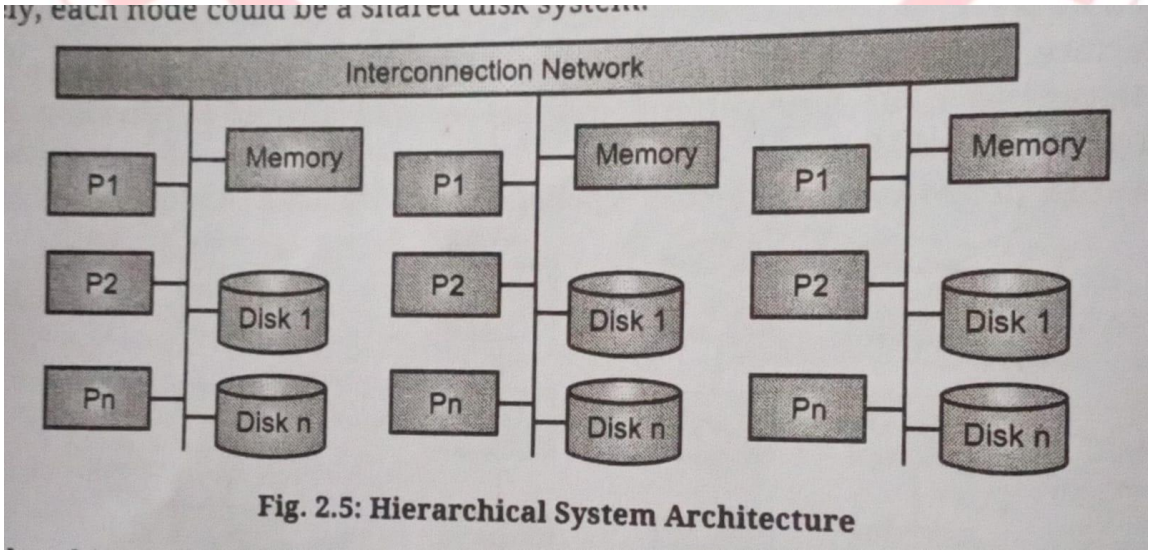


Fig. 2.5: Hierarchical System Architecture

- Hybrid (Mixed) of shared memory shard disk and shared nothing
- Combined characteristics of all 3
- Top level is shared nothing architecture
- Each node is actually a shared memory system
- Also called as NUMA- NON uniform memory architecture
- Uses local and remote memory.
- SO longer communication time

Advantages

- Improved scalability
- Minimized bottleneck
- Greater flexibility, greater performance
- Higher data availability, high extensibility

Disadvantages

- High cost
- Complex architecture

Q) Measures of performances of parallel database system

1. **Throughput**
    - Rate at which database system can process operation within given time period
    - Number of tasks that can be completed in given time interval
2. **Response time**
    - Time taken to execute specific request
    - Amount of time to complete single task from the time it is submitted
3. **Scale up**
    - Ability to keep same performance level when workload and resources increases proportionally
    - Scaleup means handling a larger task by increasing the degree of parallelism

- Scale up = Small system small problem elapsed time (single machine)/ Large system large problem elapsed time (parallel machine)

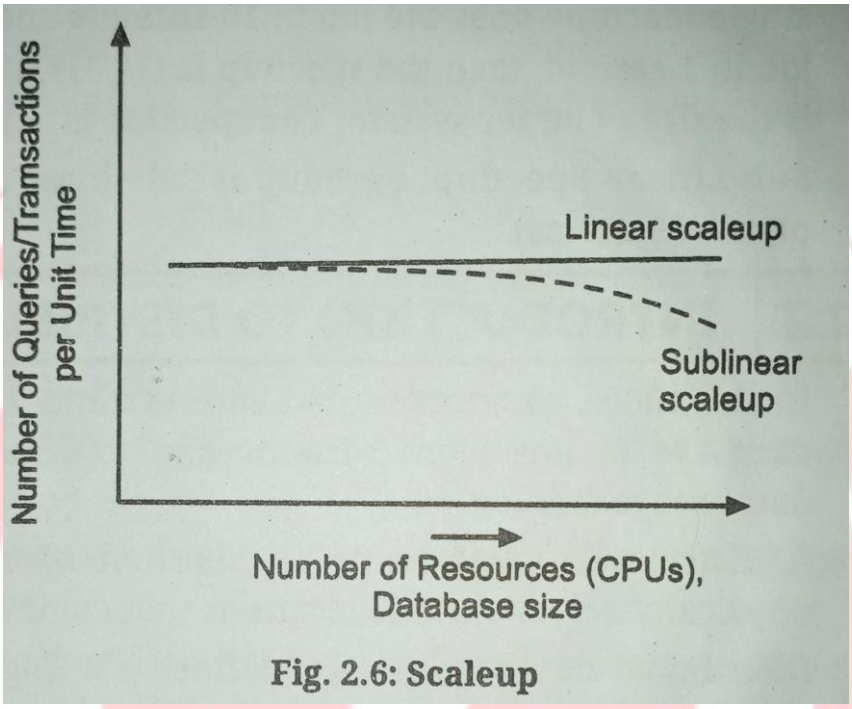


Fig. 2.6: Scaleup

a. Linear scale up
   - Scale up is linear if resources increase in proportion with problem size
   - It's the ability to maintain same level of performance when workload and resources are proportionally added.

b. Sub-linear scale up
   - If large number of problem elapsed time > small system small problem elapsed time

4. Speed up

- Ability to reduce response time of large queries by adding more resources
- Performance improvement gained as extra processing element added.
- It's like increasing the degree of parallelism
- It's measurement of performance of read only queries (data retrieval)
- Speedup = small system elapsed time / large system elapsed time

- Speed up reduces response time

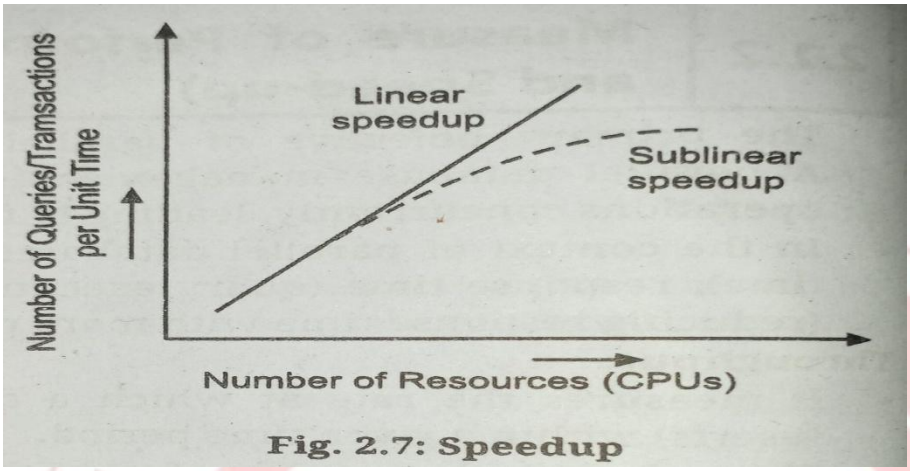


Fig. 2.7: Speedup

a. Linear speedup
   - **Linear speed = N (small system elapsed time)**
   - Single system time = 10
   - Parallel system time = 1
   - So, speed up = 10/1 = 10

b. Sub-linear speed up
   - Speedup < N

Q) What is Distributed Database?

"A distributed database is collection of interconnected databases, which are spread physically across various locations that communicate via a computer network"

OR

"Logically interrelated collection of shared data physically distributed over a computer network"

OR

"Logically related collection of shared data and a description of data is distributed over a computer network"

Distributed DBMS:

A software system that permits the management of distributed database and makes distribution transparent to the users is known as Distributed DBMS

It may be stored in same physical location or dispersed over a network of interconnected computers

Features of Distributed Database:

- Data is stored across multiple sites
- Data at each site is managed by DBMS independent of other sites
- Processors in the sites are connected via a network
- They do not have any multiprocessors configuration

Q) What is DDBMS?

DDBMS:

- Manages distributed database
- Provides mechanism to make databases transparent to users
- **"DDBMS is a centralized software system that manages DDB and provides an access mechanism that makes distribution transparent to the users"**

Features of DDBMS:

- DDBMS is used to – create, retrieve, update and delete distributed databases
- It is designed for heterogeneous database platforms
- It is used for processing large volume of data accessed by numerous users simultaneously
- DDBMS maintains confidentiality and data integrity
- It ensures data modified at any site is universally updated
- applications can access database locally and remotely

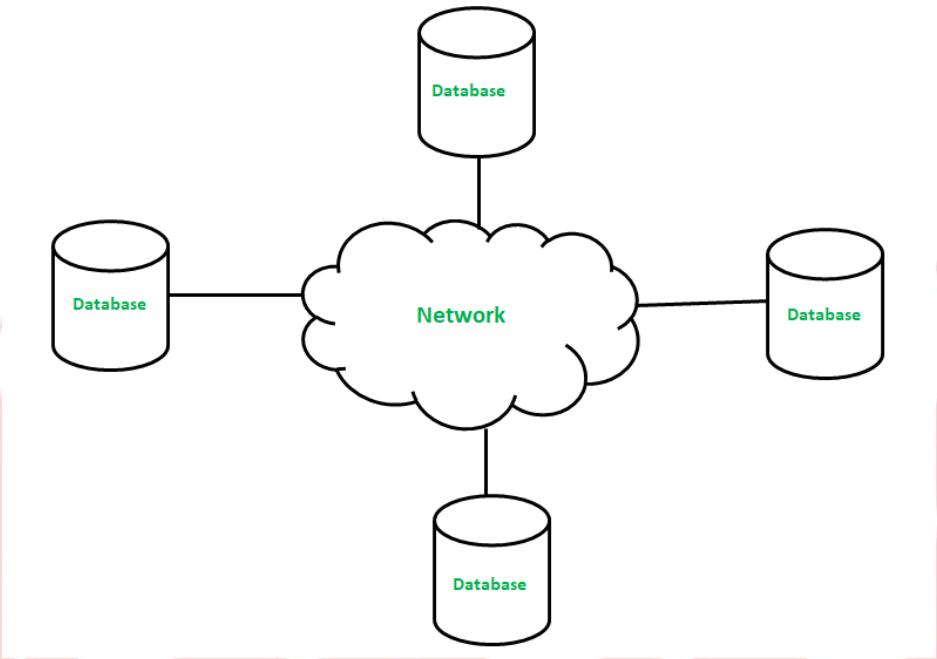
Applications in DDBMS- 2 categories:

1. **Local applications**

Requires access to local data only

2. **Global applications**

   Requires access to data from remote sites in distributed system



Q) Explain architecture of DDBMS

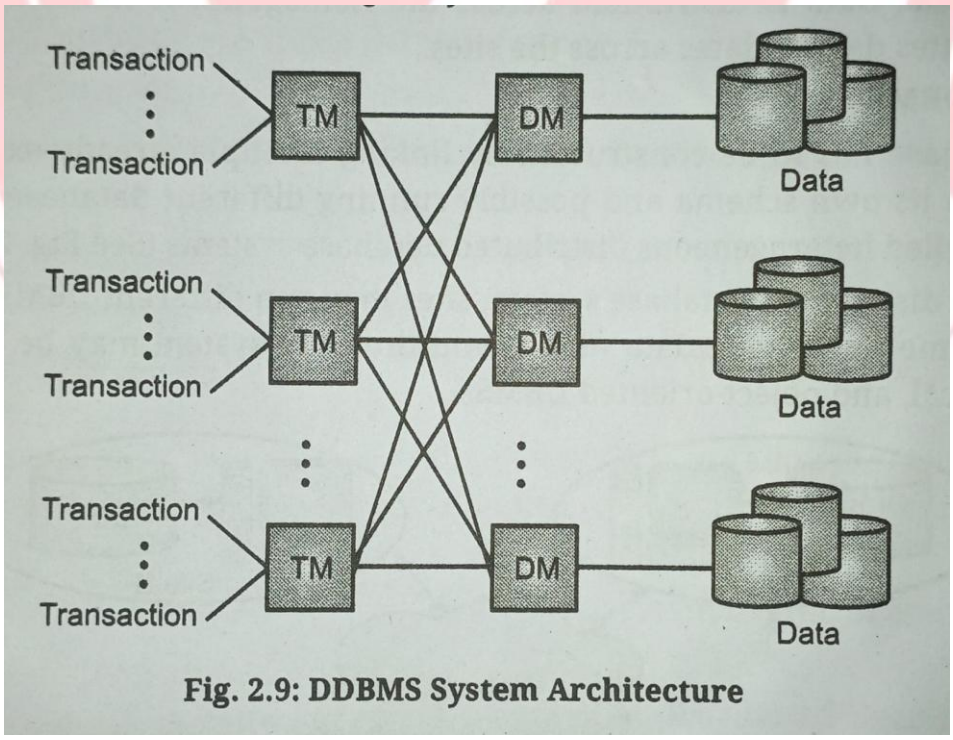


Fig. 2.9: DDBMS System Architecture

Its collection of sites interconnected by a computer network.

Components of DDBMS:

- **Computer Workstation(sites/nodes):**
  - DDBMS consists of number of computer workstations that form network system
  - Database system must be independent of computer system hardware
- **Communication Media:**
  - Communication among nodes- data transfer, information exchange→carried out by communication media
  - It's very important component of DDBMS
  - DDBMS must support several types of communication media
- **Transaction Processor (TP):**
  - Software component resides in each computer - responsible for receiving and processing both local and remote application data requests
  - Also known as AP-Application Processor OR TM- Transaction Manager
- **Data Processor (DP):**
  - Software component resides in each computer – stores and retrieves data located at that site
  - Also known as DM- Data Manager
  - In DDBMS- DP maybe centralized DBMS
- Each site can run both TM and DM
- TM-supervises interactions between users and DDBMS
- DM- manages actual DB

Q) What are different types of Distributed Database System?

DDBMS are of 2 types:

- **Homogeneous DDBMS:**
  - All sites use same DBMS product
  - All sites share common global schema

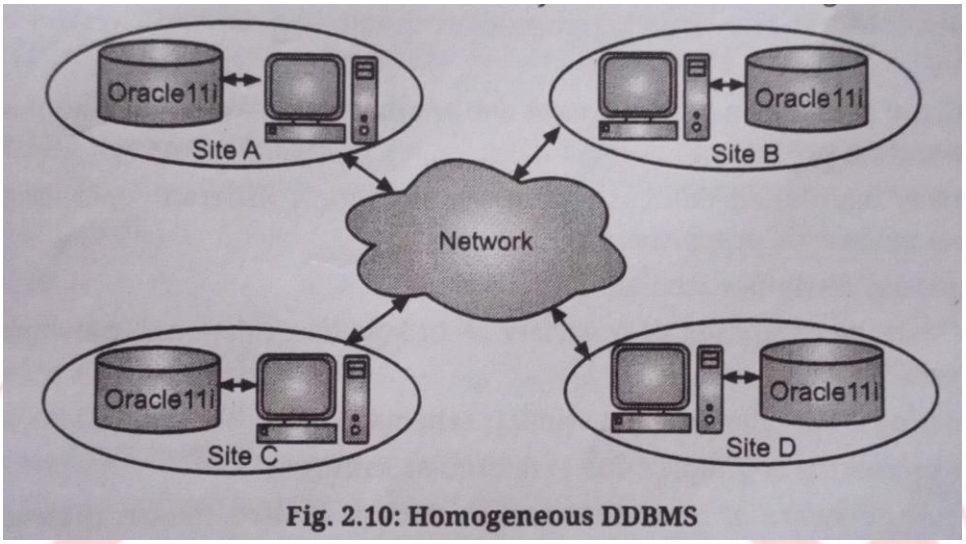
- Run identical DBMS software
- 

Fig. 2.10: Homogeneous DDBMS
- DBMS softwares on each site work together in coordination
- All sites use identical DBMS and OS
- **Properties**:
  - Each site is aware of other sites
  - Each cooperates with other sites to process user request
  - Sites use similar software
  - Database is accessed through a single interface as if it is a single database
- **2 types of Homogeneous DDB:**
  - **Autonomous:** each DB functions on its own independently
  - **Non-Autonomous:** data is distributed – central or master DBMS co-ordinates updates across sites

- **Heterogeneous DDBMS:**
  - Distributed database constructed by linking multiple already existing DB system together- with own schema but different database management software
  - Sites may run different DBMS product based on underlaying data model (relational, network, hierarchical and object-oriented DBMS)

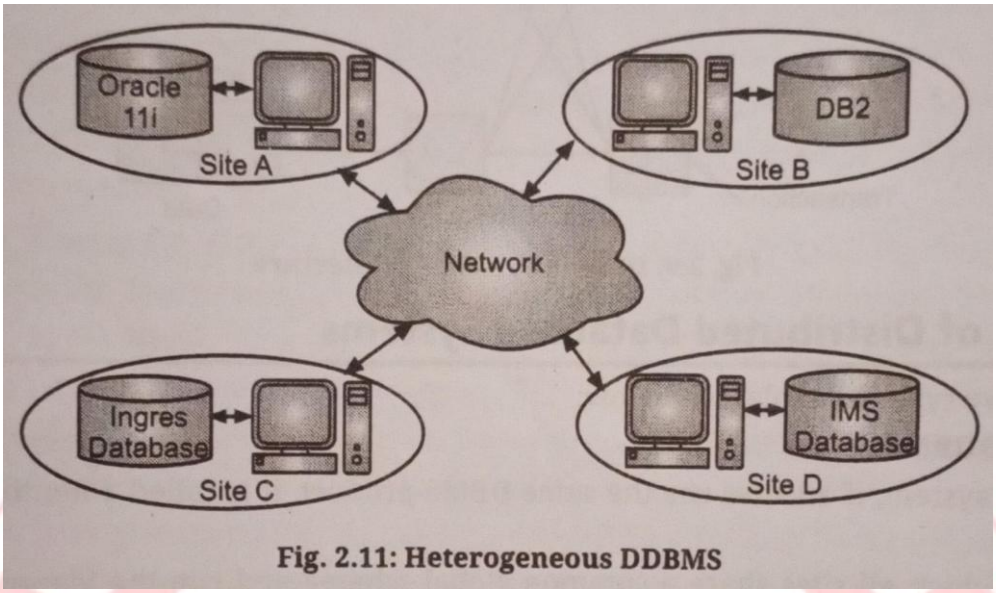


Fig. 2.11: Heterogeneous DDBMS

- Sites may not be aware about each other
- May provide limited facilities for co-operation in transaction processing
- Sites have different – OS, DBS and data models
- **Properties**:
  - Different sites use dissimilar schemas and software
  - System is composed of a variety of DBMS
  - Query processing is complex due to dissimilar schemas
  - Sites may not be aware of other sites – so less co-operation in processing user request
- **2 types of Heterogeneous Databases:**
  - **Federated**: independent in nature, integrated together to function as a single DB system
  - **Un-Federated:** DB system employ- central co-ordinating module through which DB is accessed

Q) Enlist advantages and disadvantages of Distributed Database System

Advantages:

- **Increased efficiency:** by keeping data close to most frequently used point
- **Sharing data:** can be accessed from any site
- **Improved performance:** since local data maintained locally

- **Increased accessibility:** from any site at any place
- **Increased local autonomy:** degree of control over locally stored data
- **Improved reliability/availability:** if one database fails another data is available
- **Easier expansion:** more modular, easily expandible
- **Integration of existing database:**
- **Speeding-up of Query Processing:** parallel execution of queries as data is distributed

Disadvantages:

- **Need for complex and expensive software:**
- **Data integrity:** updating data at several sites is required
- **Lack of standards:** no tools/methodologies available to convert centralized DB to distributed DB
- **Overheads for improper data distribution:**
- **Cost of development:** expensive infrastructure
- **Processing overhead:** to maintain uniformity across sites
- **Increased maintenance cost**
- **Security:** issues as data is located across several sites

Q) Differentiate between parallel DB and distributed DB

Difference between Parallel Database and Distributed Database:

Sr. No.	Characteristics	Parallel Database	Distributed Database
1.	Nature	Parallel databases are generally homogeneous in nature.	Distributed databases may be homogeneous or heterogeneous.
2.	Definition	It is software where multiple processors are used to execute and run queries in parallel.	It is software that manages multiple logically interrelated databases distributed over a computer network.
3.	Geographical location	The nodes are located at geographically same location.	The nodes are usually located at geographically different locations.
4.	Execution speed	Quicker/faster.	Slower.
5.	Overhead	Less.	More.
6.	Performance	Lower reliability and availability.	Higher reliability and availability.
7.	Scope of expansion	Difficult to expand.	Easier to expand.
8.	Backup	Backup at one site only.	Backup at multiple sites.
9.	Consistency	Maintaining consistency is easier.	Maintaining consistency is difficult.
10.	No. of Locations	Processors are tightly coupled and constitute a single database system i.e. parallel databases are centralized databases and data reside a single location.	Sites are loosely coupled and share no physical components i.e. distributed databases are geographically separated and data are distributed at several locations or sites.
11.	Query optimization	More complex.	Query optimization techniques may be different at different sites and are easy to maintain.
12.	Size of Database	Very large.	Relatively small.

Q) What is a transaction? What are the states and properties of transaction?

Transaction:

"A transaction is a unit of program execution that accesses and possibly updates various data items**"**

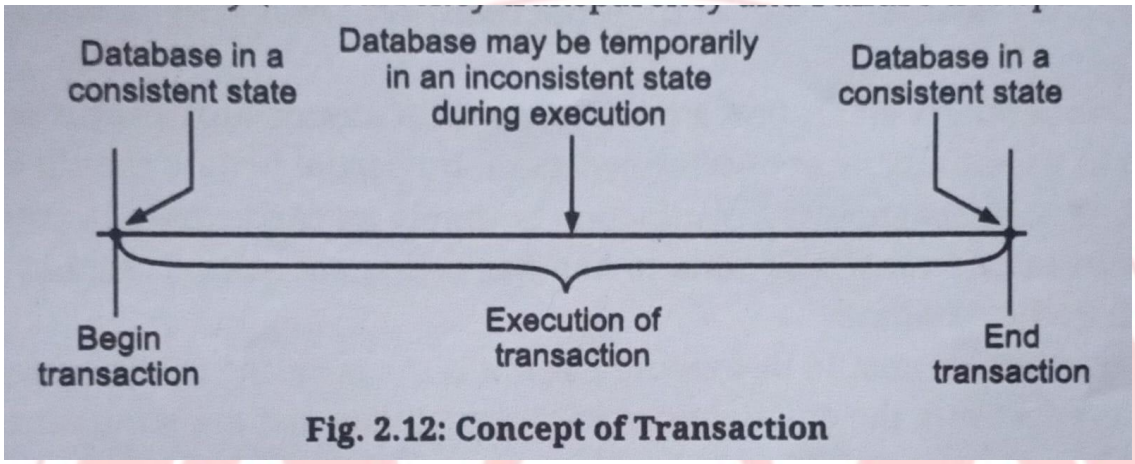
- Transaction consists of operations executed between beginning and ending of transaction
- Operations performed during transaction: insert, delete, update, retrieve
- Transaction is a collection of actions – preserving system consistency, concurrency transparency and failure transparency
- 

Fig. 2.12: Concept of Transaction

Operations performed in a transaction:

- **begin_transaction:** start of transaction execution
- **read_item or write_item:** main memory operation
- **end_transaction:** marker specifying end of transaction
- **commit:** signal specifying successful completion of transaction & will not be undone
- **rollback:** signal specifying transaction is unsuccessful and temporary changes in DB are undone.
- Committed transaction cannot be rolled back

Transaction states:

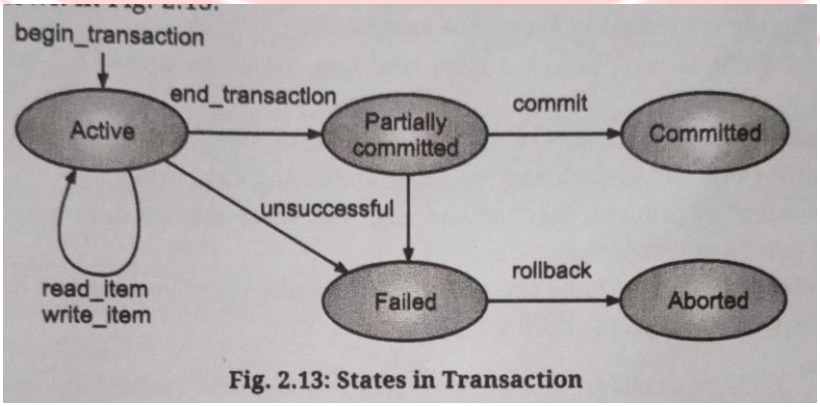


Fig. 2.13: States in Transaction

- **Active:** initial stage- transaction remains in active stage while it is executing read, write or other operations
- **Partially committed:** transaction enter in this state when the last statement has been executed
- **Committed:** transaction enters this state after successful completion of transaction and system checks have issued commit signal
- **Failed:** transaction enters failed state when normal execution cannot proceed or system checks fail
- **Aborted:** this is the state after transaction has been rolled back after failure. DB has been restored to previous state where the transaction began.

Properties of transaction:

- Transaction must maintain ACID properties
- **Atomicity:**
  - Transaction is an atomic unit of processing
  - Either it is performed entirely or not performed at all
  - No partial update should exist
- **Consistency:**
  - Its about correctness of transaction
  - Transaction should not adversely affect the database
- **Isolation:**
  - Transaction should be executed as if it is the only one in the system
  - No interference from other concurrent transaction running simultaneously
- **Durability:**
  - The change should be durable in database and not lost in case of any failure

Q) Explain transaction processing in parallel database system

- Management of database operations concurrently across multiple processors to enhance performance and scalability
- **Interquery parallelism:** each CPU executes multiple transactions simultaneously
- **Transaction dispatching:** used to manage execution of multiple transactions across different processors

- **Lock management:** to ensure data consistency and prevent conflicts when multiple transactions access same data
- **Data partitioning:** done to allow parallel processing of different parts of data
- **Concurrency control:** to ensure multiple transactions execute concurrently without conflicts and maintain data integrity
- **Recovery:** mechanism to maintain data consistency and recovery in case of failure
- **Scalability:** DB is designed to handle large volume of data across multiple processors
- Types of parallelism→
    1. **Inter-transaction parallelism**
    2. **Data parallelism**
- **Inter-transaction parallelism:**
- Multiple independent transactions executed concurrently in parallel by each CPU
- Also known as parallel transaction processing
- "Parallelism among queries"
- It helps to scale up transaction processing
- Supports large number transactions per second

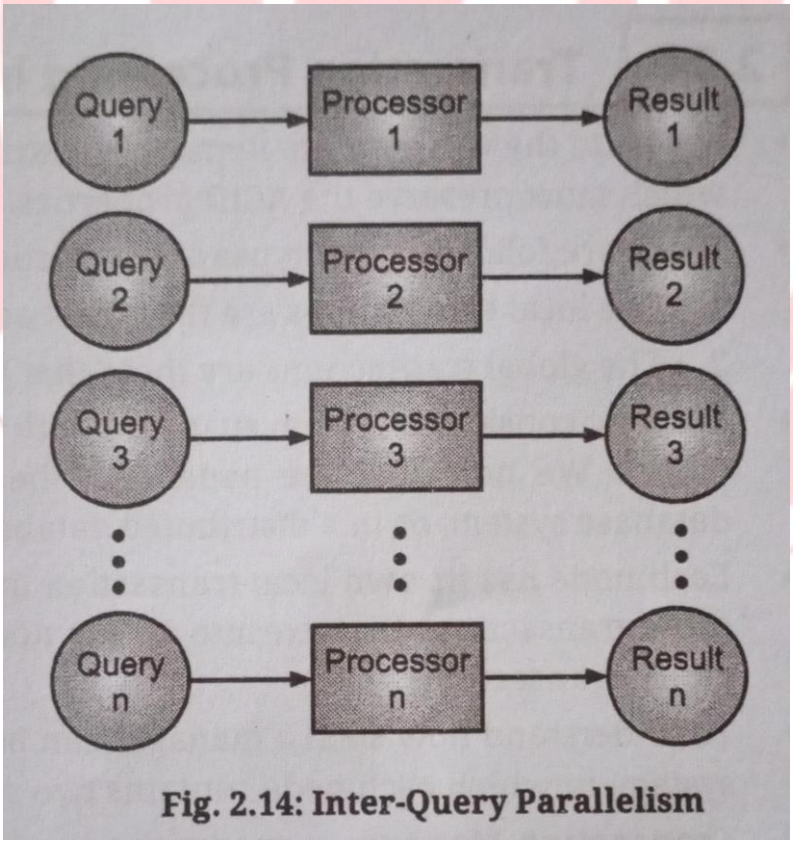


Fig. 2.14: Inter-Query Parallelism

- **Advantages:**

  - Increases transaction throughput
  - Scales up transaction processing to large number per second
  - It is easiest form of parallelism particularly shared memory parallelism
- **Disadvantages:**
  - It is complicated to support shared disk and shared nothing architecture
  - Response time of each transaction is almost same as if running in isolation
  - Does not help speeding up long running queries- as each query run sequentially
- **Data parallelism:**
- "A large task is divided into smaller, independent subtasks that are processed concurrently by different processors on the same data"
- **Advantages:**
  - **Increased performance:** reduces query execution time and improves system responsiveness
  - **Scalability**: parallel DB can accommodate growing DB and increasing transaction loads
  - **High availability:** data replication ensures data availability
  - **Efficient resource utilization:** better utilization of CPU and I/O resources

Q) what are the benefits and challenges of transaction processing?

- **Benefits→**
- **Increased throughput:** as transactions are executed faster
- **Reduced response time:** as transaction run faster
- **Improved scalability:** can handle larger workloads
- **Enhanced availability:** data is replicated at different nodes- in case of failure of one node system ensures high availability at another node
- **Challenges→**
- **Complexity:** management of distributed environment is complex
- **Inter-node communication:** introduces overhead
- **Load balancing:** even load distribution is crucial

Q) Explain transaction processing system in distributed database system

2 types of transactions:

Local Transactions: - access and update data only one local database

Global Transaction: - access and update data in several local databases

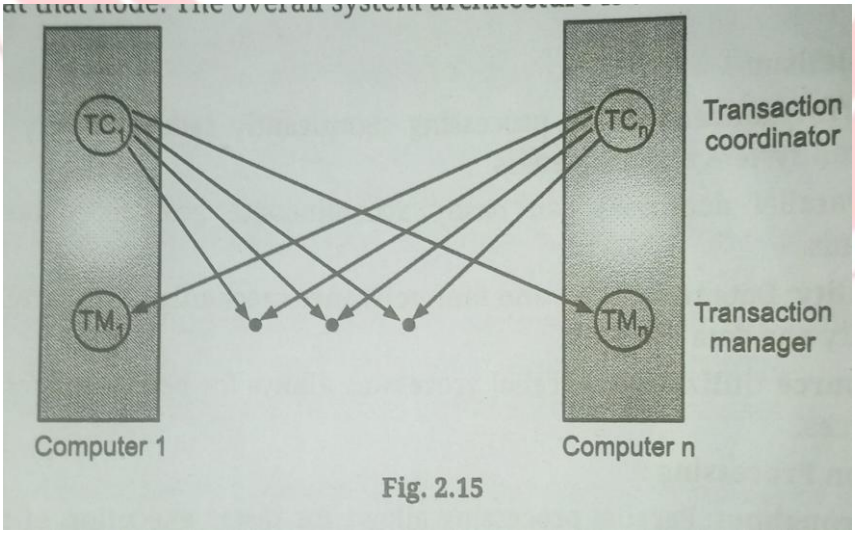
- Local transaction manager at local nodes
- And various transaction managers to handle global transactions
- At each node there are 2 subsystems:
- **Transaction Manager (TM):**
    - Manages transactions that access data stored at the node.
    - Transaction may be local or global
    - TM maintains logs for recovery purposes
- **Transaction Co-ordinator (TC):**
    - Co-ordinates execution of various transactions initiated at the node
    - Local or global
    - TC is responsible for
        - Starting execution of transaction
        - Breaking transaction into sub transactions and distributing them to appropriate nodes
        - Coordinating termination of transactions
    - 

Fig. 2.15
- **Protocols used for Transaction Processing in DDBMS:**

1. **Two-Phase Commit Protocol(2PC):**
    - **2 phases:**
    - **Prepare phase-** all participants agree to commit or abort the transaction
    - **Commit phase-** decision is executed synchronously across all participants
    - 2PC ensures that all participants agree to either commit or rollback a transaction
    - Ensures data consistency across entire system
    - Prepare: asks all nodes if they are ready to commit
    - Each node replies “yes” or “no”
    - If “no” or node fails to respond within timeframe- rollback the transaction
    - If all positive responses- commit message sent to nodes and waits for acknowledgement before finalizing the commit operation

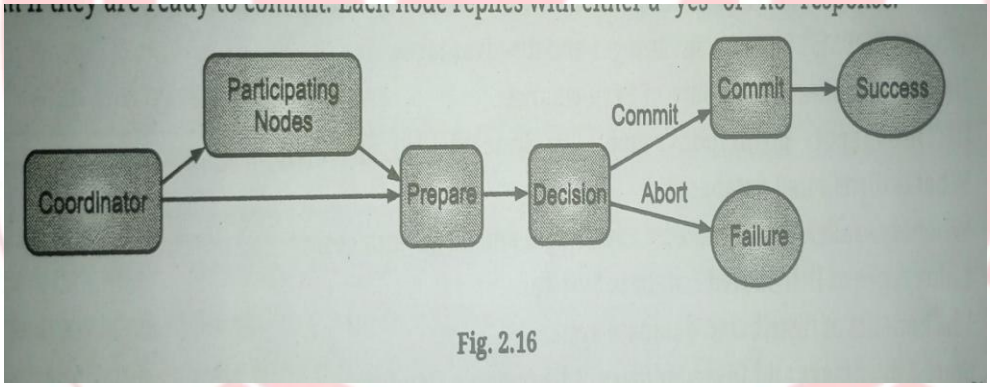


Fig. 2.16

-

1. **Three-Phase Commit Protocol(3PC):**
    - Extends 2PC by adding extra phase
    - Pre-commit phase
    - Addresses certain failure scenarios- blocking indefinite blocking in 2PC
    - In 3PC- participants agree to prepare to commit- then to commit -and finally to complete or abort the transaction
    - 3PC reduces the risk of blocking in 2PC

    - Introducing additional decision-making phase
- **Advantages of distributed transactions:**
    - **High Availability:** multiple sites assure high availability of data
    - **Data Consistency:** operations are performed in coordination so data remains consistence
    - **Scalability:** DB can handle large transaction on data stores at different sites
    - **Fault Tolerance:** distributed system can handle faults and ensures proper transaction. In case of failure roll back and use alternate resources
    - **Data Redundancy:** copies of data at different sites – easy recovery at the time of failure
    - **Guarantees Transaction:** durability and isolation make sure if any transaction is committed – changes last even if failure occur
    - **Improved performance:** concurrent execution of transactions increases system speed and efficiency
- **Challenges:**
    - **Security risks:** risk of security breach at different sites
    - **Complexity:** complex to implement and maintain data at various sites