Advanced Database Concepts

Chapter - 1: Introduction to Advanced Databases

Q.1 Write about CENTRALIZED DATABASE ARCHITECTURE

Answer:

1. In Centralized Architecture, the mainframe computers/systems are used to do the processing work including user application programs and user interface programs as well as DBMS functionalities.
2. Earlier users accessed such systems via **Computer Terminals.**
3. These **computer terminals** used to have only **display** capability.
4. Therefore, processing is done in this **mainframe systems** and the sent to **computer terminals** to display the outcome.
5. These **display terminals** and **mainframe systems** are connected via various kinds of network.

Q.2 Explain 1-tier, 2-tier and 3-tier architecture

Answer:

A) 1-Tier Architecture:

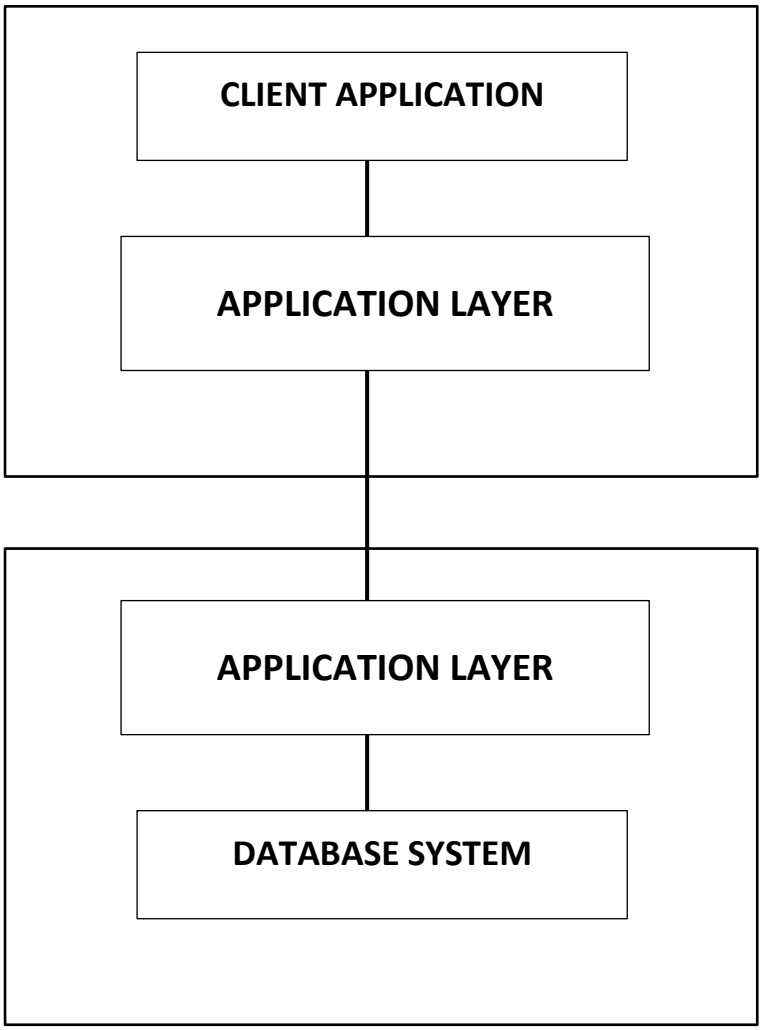
- In this type of architecture, the actual database is directly available to the end users.
- It means that users can directly sit on the DBMS and use it.
- Any changes done here by the user will be done on the actual database itself.
- This architecture does not provide handy tools to the end users.
- Such architecture is useful to build **local applications** where programmers need to communicate with the database for the quick responses.
- A simple one tier architecture example would be anytime you install a Database in your system and access it to practice SQL queries.

B) 2-Tier Architecture:

- This architecture is same as client – server architecture.
- In this architecture, applications on the client side can directly communicate with the database stored at the server side.
- For such communication to happen, API's such as **ODBC, JDBC** are used.
- The user interfaces and applications are run on the client side
- The server side is responsible to provide functionalities such as: **query processing and transaction management.**
- To communicate with DBMS at the server side, client side application establishes a connection with the server side.
- Example: A Contact Management System created using MS- Access.

C) 3-Tier Architecture:

- This architecture is the most popular and most used type of architecture in the real world.
- This is an extension of 2-tier architecture.
- In 3-tier architecture, other than client-side and server-side there is an third layer between client and the server side which is known as **Application Layer (Business Logic Layer).**
- This layer is used to carry out the communication between client side application and server side database system.
- It takes the request from the client application and further sends it to the server side system to process and again takes the output from the server side and sends it to the client application to display it.
- Client end user has no idea about the existence of the database beyond application layer and server side database has no idea about the existence of the client side application.
- This architecture is more secured than 1-tier and 2-tier architecture since DBMS is less exposed to the end user.
- Such architecture is used in case of large web application.



Chapter - 2: Parallel Databases

Q.1 Explain parallelism in database

Answer:

1. Data can be partitioned on multiple disks for parallel processing.
2. Different relational queries such as join, sort, aggregation can be done in parallel manner.
   ➔ Data can be partitioned and multiple processors can work independently on its own partition.
3. Queries are expressed in high level language
   ➔ Which makes parallelization easier.
4. Different queries can be executed at the same time in parallel manner and the concurrency control mechanism will take care of conflicts.
5. Thus, database naturally lend themselves to parallelism.

Q.2 Explain need of Parallel database

Answer:

1. Parallel machines are becoming quite common and affordable
2. Prices of microprocessors, memory and disks have dropped sharply.
3. Recent desktop computers feature multiple processors and this trend is projected to accelerate.
4. Databases are growing increasingly large.
5. Large volumes of transaction data are collected and stored for later analysis.
6. Multimedia objects like images are increasingly stored in databases.
7. To improve the performance of such a large database, parallel database is needed.
8. Such large amount of data cannot be accessed and used from multiple locations faster and efficiently if stored at a single location.
9. Companies now-a-days have branches in multiple cities across the world, to access data from every branch parallel database system is needed.

Q.3 Explain goals of Parallel Databases

Answer:

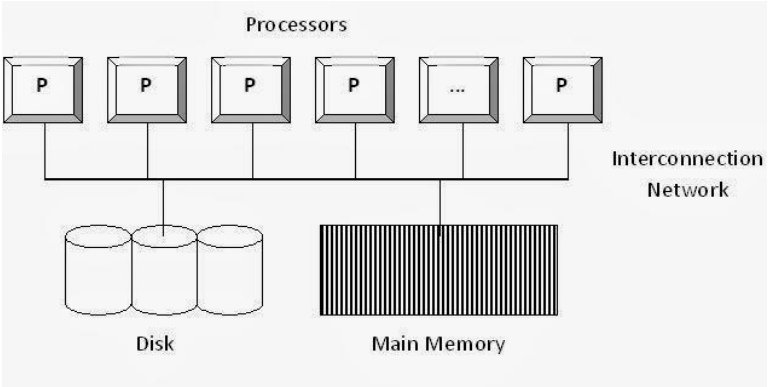
1. **Improve the performance:**
   ➔ The performance of the system can be improved by connecting multiple CPUs and disks in parallel.
   ➔ Many small processors can also be connected in parallel
2. **Improve availability of the data:**
   ➔ Data can be copied to multiple locations to improve the availability of the data across the world

3. **Improve reliability:**
   ➔ Reliability of the system is improved with completeness, accuracy and availability of data.
4. **Provide distributed access of data:**
   ➔ Companies have their branches in multiple cities so they can access the data at those branches with help of parallel database system.

Q.4 Explain different parallel database architectures

Answer:

1. **SHARED MEMORY ARCHITECTURE:**
   A) In this architecture, multiple processors share single memory space.
   B) Several processors are connected to the main memory and disk setup via interconnection network.
   C) This interconnection network is usually a high speed network such as bus, mesh and hypercube, which makes data sharing easier among various components such as processors, memory and disk, in the network.
   D) **Advantages:**
      a) Simple in implementation
      b) Establishes effective connection between the processors through single memory addresses space.
      c) Above point leads to less communication overhead.
   E) **Disadvantages:**
      a) Higher degree of parallelism cannot be achieved because multiple processors access single memory via same interconnection network.
      b) The above causes bottleneck in the interconnection network, especially in bus interconnection network.
      c) Adding more processors may affect the working speed of other processors.
      d) Cache-coherency should to be maintained, which means if a processor tries to read the data which is used and modified by other processor, then it should ensure that the data is of latest version.
      e) Degree of parallelism is limited since adding more processors can make other processors slower.



2. SHARED DISK ARCHITECTURE:

A) Single disk or disk setup is shared among all the available processors.

B) Also these processors have their own private memory.

C) **Advantages:**

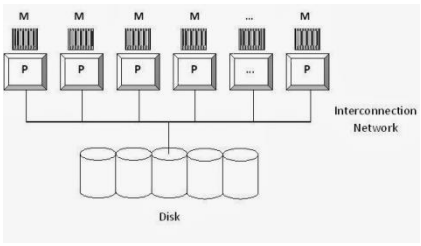
a) Failure one processor would not cause the entire system to stop working.

b) Supports larger number of processors as compared to shared disk architecture.

D) **Disadvantages:**

a) Complex to implement as compared to shared disk architecture.

b) Inter-processor communication is slower. Because each processor has its own private memory and if one processor need to access memory of other processor then it needs additional software support.



3. SHARED NOTHING ARCHITECTURE:

A) In this architecture every processor has its own memory and disk setup.

B) It can be considered as multiple individual computers connected through a high speed interconnection network using regular network protocols and switches.

C) This type of architecture is used in **Distributed Database Systems**.

D) In this type of architecture, it insists to have similar type of systems in the network which are called **Homogenous systems**.

E) **Advantages:**

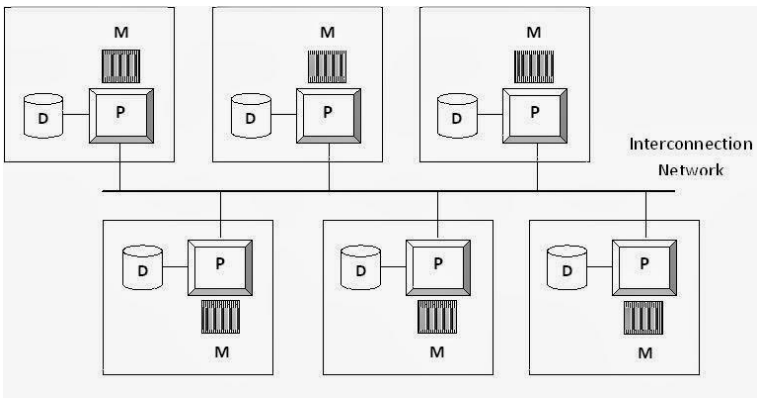
a) Number of processors in the network is scalable, that is, the design is flexible to add or remove one or more number of processors.

b) Unlike the other two architectures, if the data is not available at the local processors then only the data request is to be forwarded through the interconnection network.

F) **Disadvantages:**

a) Non-local disk access is costly, that is, if the requested data is not available at the local processor, then the request should to be routed towards the processor where data is available, which is slightly complex to implement.

b) Communication cost of transporting the data among the processors is higher.



Q.5 Explain response time, speed up and scale up

Answer:

1. **Response Time:** It is the time required to complete a given single task.
2. **Speed Up:** It is the process of increasing the degree of parallelism to complete a given task and reducing the response time. The required for running a task is inversely proportional to the number of resources.

   Example:

   a) 10 seconds to scan 10,000 records using 1 CPU

   b) 1 second to scan 10,000 records using 10 CPUs
3. **Scale Up:** It is the ability to keep performance constant, when number of processes and number of resources increases proportionally.

   Example:

   A) 1 second to scan 1,000 records using 1 CPU

   B) 1 second to scan 10,000 records using 10 CPUs

Q.6 Explain different types parallelism in DBMS

Answer:

Parallelism in a query allows us to execute multiple queries in parallel by decomposing them in to smaller parts that work in parallel. We can achieve parallelism in a query by the following methods:

1. **I/O Parallelism:**

- It is a form of parallelism in which relations are partitioned on to the multiple disks with a motive to reduce the retrieval time of the relations from the disk.
- Within, the data inputted is partitioned and then processing is done with each partition.
- The results are merged after processing all the partitioned data.
- It is also known as **data partitioning.**
- It can be achieved in four ways as follows:

A) Hash Partitioning:

➔ As we know that 'hash' function is fast, mathematical function.

➔ Each row of the relation is hashed on to the partitioning attributes.

➔ For example: We have Disk1, Disk2, Disk3 and Disk4, if the hash function returns 3, then the row will be placed in Disk no. 3

B) Range Partitioning:

➔ In range partitioning, it issues continuous attribute value ranges to each disk.

➔ For example: If we have Disk0, Disk1 and Disk2, then we may assign relation with value less than 5 to Disk0, relation with value between 5-40 to Disk1 and relation with value greater than 40 to Disk2.

C) Round-robin Partitioning:

➔ In this partitioning relations are studied in any other.

➔ The ith tuple is sent to disk number (i%n)

➔ So each disk takes turn while receiving new tuples

➔ It ensures even distribution of the tuples across the disks.

➔ It is ideal for an application that wish to read the entire relation sequentially for each query.

2. **Interquery Parallelism:**
   A) In this parallelism different queries are allowed to be executed on multiple processors in parallel.
   B) Pipelined parallelism can be achieved by using this type of parallelism, which improves the output of the system.
   C) For example:
   6 queries, each takes time of 3 seconds so total time = 18 seconds, but this parallelism makes the total time = 3 seconds.
   D) Main aim is to scale up the processing of the system. However, it is difficult to achieve this parallelism every time.
3. **Intraquery Parallelism:**
   A) In this parallelism, single query is divided into sub queries and are executed on multiple processors in parallel.
   B) It improves the evaluation time of the query.
   C) It also reduces the response time of the system.
   D) Main aim is to speed up the long running queries.
   E) It can be obtained in two ways as follows:

➔ **Intraoperation:**

In this, we parallelize the execution of each individual operation of a single query such as sort, joins, projection and so on.

The degree of parallelism is very high in this type.

It is natural in database systems.

For example:

SELECT * FROM Vehicles ORDER BY Model_Number;

In the above query, relational operation is sorting and the relation can have large amount of records in it. So we can apply the sorting on different subsets of the relation which will reduce the time to sort the data.

➔ **Interopertion:**

When different opertions in a query expression are executed in parallel, then it is called interoperation parallelism.

They are of two types:

- **Pipelined:**

  The output row of first operation is consumed by the second operation even before the first operation has produced entire set of rows in its output. It is useful for smaller number of CPUs and avoids writing of intermediate results on the disk.

- **Independent:**

  The operations in a query that are not dependent on each other can be executed in parallel. This is very useful in the case of lower degree of parallelism

Chapter 3 – Distributed Databases

Q.1 Define what is Distributed Database

Answer:

It is a collection of multiple interconnected databases, which are stored across multiple locations in the world that communicate via a computer network.

Q.2 Write features of distributed databases

Answer:

1. The databases in the collection are logically interrelated to each other. Often they represent a single logical database.
2. Data is stored at multiple locations. Data in each site can be managed / accessed by a DBMS independent of other sites.
3. It is not a loosely coupled database.
4. The processors at the site are not connected via a network. They do not have multi-processor configuration.
5. It incorporates transaction processing, but it is not similar to transaction processing system.

Q.3 Write what is Distributed DBMS

Answer:

It is a centralized software system that manages a distributed database in a manner as if they all are stored at a single location. It synchronizes the data periodically and ensures that the data modification / deletion done at one location will be reflected on the data stored at another location.

Q.4 Features of DDBMS

Answer:

1. It is used to create, retrieve, update and delete the distributed databases.
2. It synchronizes the database periodically and provides accessing mechanisms by virtue of which distribution becomes transparent to the users.
3. It ensures that data modification done at one location is universally updated.
4. It is used in the application areas where large volumes of data is processed and accessed by numerous users simultaneously.
5. It is designed for heterogeneous database platforms.
6. It also maintains confidentiality and data integrity of the databases.

Q.5 Write down the goals of DDBMS

Answer:

1. **Reliability:**
   - Distributed database management systems is more reliable because if any connected computer system is failed to perform then other computer systems can complete the task without any delay.
2. **Availability:**
   - In DDBMS if any computer server fails to perform and stopped working in any time then other computer servers can perform the task that is requested.

3. **Performance:**
   - The data and information can be accessed from DDBMS from different locations. It is easy to handle and maintain.

Q.6 Different types of DDBMS

Answer:

1. **Homogenous:**

- It is a collection or a network of identical databases.
- All the databases in this network stores data identically.
- OS, DDBMS and data structures used – all are same at all the sites which makes it easy to maintain and manage.
- It is much simpler to design, manage and add a new site in the network.
- It also helps to improve the parallel processing capabilities of multiple sites.
- It has two types:
  A) **Autonomous:**
  It contains the databases which are independent and functions on its own.
  They are integrated by a controlling application and uses message passing to communicate the data updates.
  B) **Non-Autonomous:**
  Data is distributed across the homogenous nodes and is controlled by a central or master DBMS which co-ordinates the data updates across the sites.

2. **Heterogeneous:**

- It is opposite of Homogenous system.
- So every site in this network has different OS, different DDBMS and different models causing it difficult to manage.
- In this system, one site can be completely unaware of the other sites.
- It causes limited cooperation in processing the user request, so for this translations are used to establish communication between the sites.
- Its designing is hard because a system must provide interoperability between different sites which might be using different database management software and hardware.
- It has two types:
  A) **Federated:**
  The heterogeneous database systems are independent of each other and are integrated together so that they can function as a single system.
  B) **Un-Federated:**
  The heterogeneous database systems employ a central controlling module through which the databases are accessed.

Q.7 Data Replication in DDBMS

Answer:

1. The same data is stored at more than one sites
2. This ensures the availability of data because even if one site goes down, data will be available at other site(s).
3. It also provides faster access and reduces the need for data to be travelled on the network hence improving the performance of the system.
4. Types of data replication techniques:

- Snapshot replication
- Near-real time replication
- Pull replication

5. **Advantages:**
   a) **Reliability:** In case of failure of any site, the system won't stop working because The copy of the data is available at other site(s)
   b) **Reduction in Network load:** Since local copies of the data are available at multiple sites, query processing can be done with network load and in the prime hours and data updates can be done in non-prime hours.
   c) **Quicker Response:** The availability of local copies of data ensures quick query processing and consequently quick response time.
   d) **Simpler transaction:** Since data is locally available, the transactions would need less number of table joins from different sites and minimal coordination between different sites.
6. **Disadvantages:**
   a) **Increased storage requirement:** Since multiple copies of a single data are to be stored it will cause the need of more space and increase in storage costs.
   b) **Increased cost and complexity of data updating:** Each time an item is updated, the update needs to be reflected at all the copies stored at different sites. This will need complex synchronization techniques and protocols.
   c) **Undesirable application – database coupling:** If complex update mechanisms are not provided, then complex coordination at application level will be required resulting in undesired application-database coupling.

Q.8 Data Fragmentation in DDBMS

Answer:

1. The system fragments the relations into smaller parts and stored them at different sites in the system.
2. Usually fragments are stored as per there usage at the site.
3. When a query is made, first local database at the site is checked and if not found at local site then other sites are checked.
4. Fragmentation is a process of dividing the table into set of smaller tables. These subsets are called **fragments**

5. Fragmentation are of three types: Vertical, Horizontal and Hybrid Fragmentation
6. Fragmentation should be done such a manner that original table can be reconstructed from fragments.
7. This rule or requirement is called **re-constructiveness**
8. **Advantages:**

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

9. **Disadvantages:**

- Since data which is needed cannot be available at the local site, the accessing time of the data from other sites can be slower.
- In case re-constructiveness, complex and expensive techniques are needed.
- Lack of back-up copies of data in different sites may render the database ineffective in case of failure of a site.

10. **Vertical Fragmentation:**

- Fragments are generated by grouping the fields or columns of the table.
- The re-constructiveness is maintained by keeping primary key of the table in every fragment.
- This fragmentation is useful for data security and privacy.
- For example:
  DATABASE: STUDENT -> |Regd_no | Name | Course | Address | Semester | Fees | Marks
  FRAGMENT (FOR ACCOUNT SECTION): CREATE TABLE STUD_FEES AS SELECT Regd_no, Fees FROM STUDENT;

11. **Horizontal Fragmentation:**

- Fragments are generated by grouping the tuples of the table according to the values of one or more fields.
- The re-constructiveness is maintained by keeping all the columns of the table in every fragment.
- For example:
  DATABASE: STUDENT -> |Regd_no | Name | Course | Address | Semester | Fees | Marks
  FRAGMENT (FOR CS Students): CREATE TABLE STUD_CS AS SELECT * FROM STUDENT;

12. **Hybrid Fragmentation:**

- Combination of both vertical and horizontal fragmentation
- It is flexible since it generates fragments with minimal extra information
- However reconstruction of the original table is often an expensive task.
- It can be achieved in two ways:
  - At first, generate set of horizontal fragments, and then generate vertical fragments from one or more of the horizontal fragments.

- At first, generate set of vertical fragments, and then generate horizontal fragments from one or more of the vertical fragments.

Q.9 Write Concurrency Control Methods in DDBMS

Answer:

1. **Lock Based Protocol:**

- This lock is applied to avoid concurrency problems in such a way that the lock is applied on one transaction and other transaction can access it only when the lock is released.
- The lock is applied on write and read operations.
- It is an important protocol to avoid deadlock.

2. **Shared Lock Protocol:**

- The transaction can activate shared lock on the data to read its content.
- The lock is shared in such a way that any other transaction can activate the shared lock on the same data for reading purpose.

3. **Exclusive Lock Protocol:**

- The transaction can activate exclusive lock on a data for read or write operations.
- In this system, no other transaction can activate the any kind of lock on that same data.

Q.10 Write about Two-phase commit protocol in detail.

Answer:

1. It is an atomic commitment protocol
2. It is a distributed algorithm which can coordinate all the processes that participate in the database and decide to commit or terminate the transaction.
3. This protocol is based on commit and terminate action
4. This protocol ensures that all the processes that accesses the database server can receive and implement the same action (Commit or Terminate) when a local network failure occurs.
5. It provides automatic recovery mechanism when system failure occurs.
6. The location at which original transaction takes place is called **coordinator** and the where the sub processes take place is called as **Cohort.**
7. **Commit Request:**

- In this phase coordinator attempts to prepare all the cohorts and take necessary steps to commit or terminate the transaction.

8. **Commit Phase:**

- This phase is based on voting of cohorts and coordinator decides to commit or terminate the transaction accordingly.