Introduction to Parallel Databases

Companies need to handle huge amount of data with high data transfer rate. The client server and centralized system is not much efficient. The need to improve the efficiency gave birth to the concept of Parallel Databases.

Parallel database system improves performance of data processing using multiple resources in parallel, like multiple CPU and disks are used parallely.

It also performs many parallelization operations like, data loading and query processing.

Goals of Parallel Databases

The concept of Parallel Database was built with a goal to:

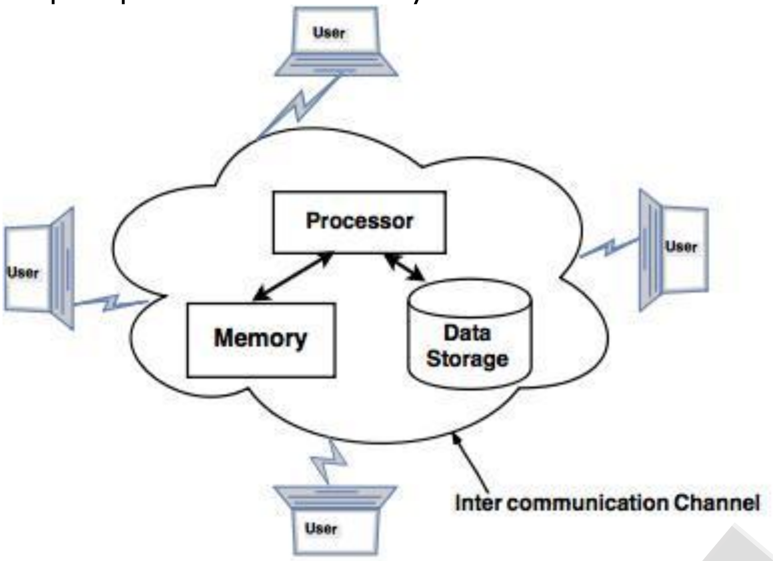
Improve performance:
The performance of the system can be improved by connecting multiple CPU and disks in parallel. Many small processors can also be connected in parallel.

Improve availability of data:
Data can be copied to multiple locations to improve the availability of data.
For example: if a module contains a relation (table in database) which is unavailable then it is important to make it available from another module.

Improve reliability:
Reliability of system is improved with completeness, accuracy and availability of data.

Provide distributed access of data:
Companies having many branches in multiple cities can access data with the

help of parallel database system.

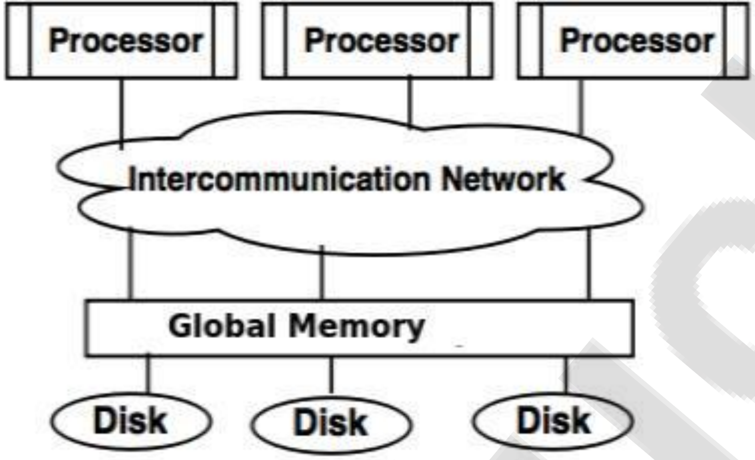


Parallel database system

Types of Parallel Database Architecture

Shared memory system

- Shared memory system uses multiple processors which is attached to a global shared memory via intercommunication channel or communication bus.
- Shared memory system have large amount of cache memory at each processors, so referencing of the shared memory is avoided.
- If a processor performs a write operation to memory location, the data should be updated or removed from that location.



Shared Memory System in Parallel Databases

Advantages of Shared memory system

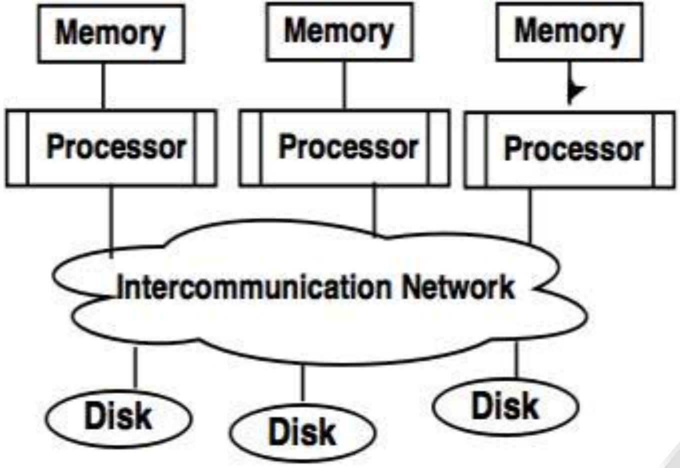
- Data is easily accessible to any processor.
- One processor can send message to other efficiently.

Disadvantages of Shared memory system

- Waiting time of processors is increased due to more number of processors.
- Bandwidth problem.

Shared Disk System

- Shared disk system uses multiple processors which are accessible to multiple disks via intercommunication channel and every processor has local memory.
- Each processor has its own memory so the data sharing is efficient.
- The system built around this system are called as clusters.



Shared disk system in Parallel Databases

Advantages of Shared Disk System

- Fault tolerance is achieved using shared disk system.

  Fault tolerance: If a processor or its memory fails, the other processor can complete the task. This is called as fault tolerance.

Disadvantage of Shared Disk System

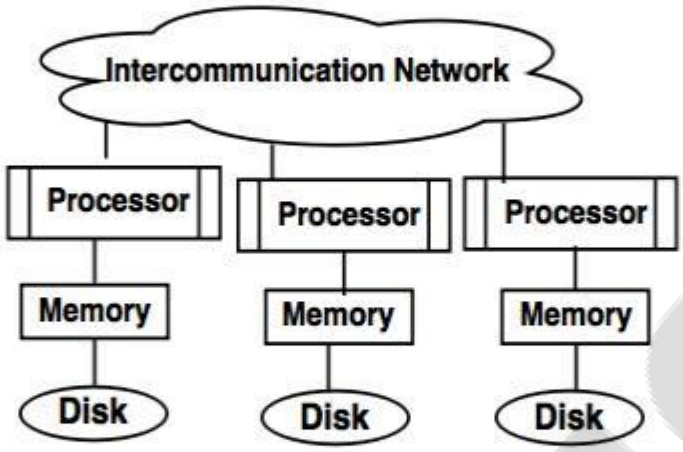
- Shared disk system has limited scalability as large amount of data travels through the interconnection channel.
- If more processors are added the existing processors are slowed down.

Applications of Shared Disk System

Digital Equipment Corporation(DEC): DEC cluster running relational databases use the shared disk system and now owned by Oracle.

Shared nothing disk system

- Each processor in the shared nothing system has its own local memory and local disk.
- Processors can communicate with each other through intercommunication channel.
- Any processor can act as a server to serve the data which is stored on local disk.



Shared nothing disk system in Parallel Databases

Advantages of Shared nothing disk system

- Number of processors and disk can be connected as per the requirement in share nothing disk system.
- Shared nothing disk system can support for many processor, which makes the system more scalable.

Disadvantages of Shared nothing disk system

- Data partitioning is required in shared nothing disk system.
- Cost of communication for accessing local disk is much higher.

Applications of Shared nothing disk system

- Tera data database machine.
- The Grace and Gamma research prototypes.

Hierarchical System or Non-Uniform Memory Architecture

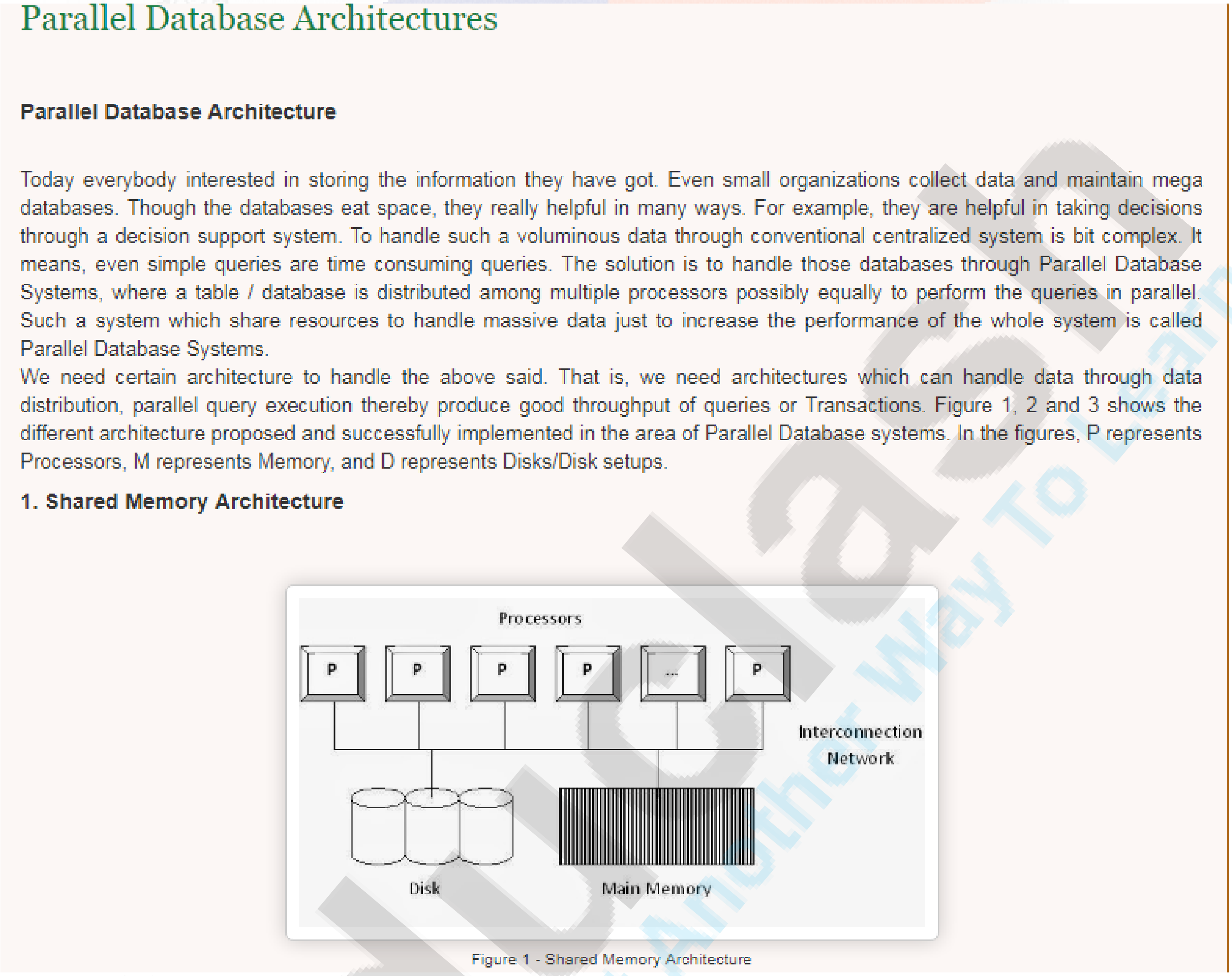
- Hierarchical model system is a hybrid of shared memory system, shared disk system and shared nothing system.
- Hierarchical model is also known as **Non-Uniform Memory Architecture (NUMA).**
- In this system each group of processor has a local memory. But processors from other groups can access memory which is associated with the other group in coherent.
- **NUMA** uses local and remote memory(Memory from other group), hence it will take longer time to communicate with each other.

Advantages of NUMA

- Improves the scalability of the system.
- Memory bottleneck(shortage of memory) problem is minimized in this architecture.

Disadvantages of NUMA

The cost of the architecture is higher compared to other architectures.



Parallel Database Architectures

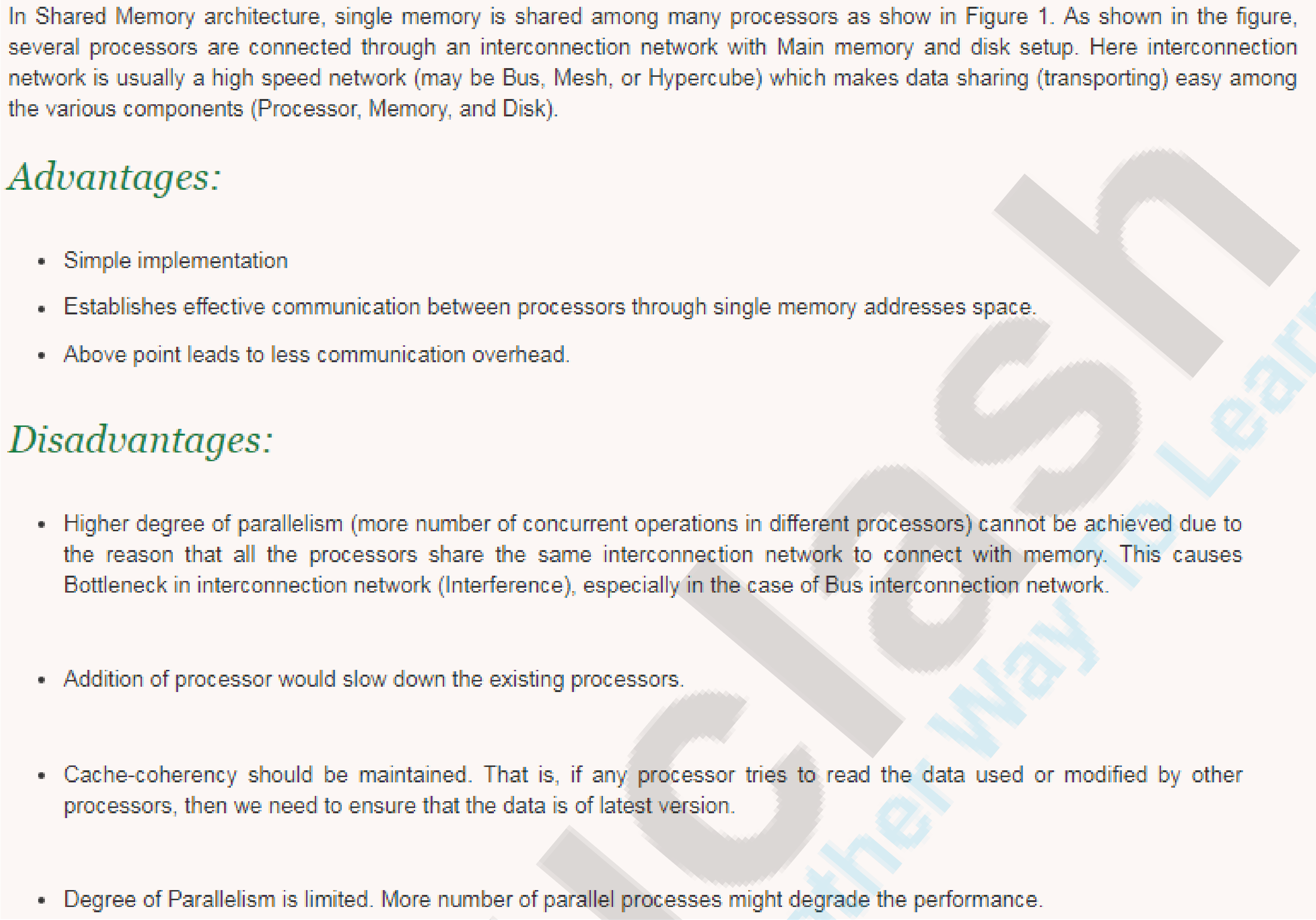
Parallel Database Architecture

Today everybody interested in storing the information they have got. Even small organizations collect data and maintain mega databases. Though the databases eat space, they really helpful in many ways. For example, they are helpful in taking decisions through a decision support system. To handle such a voluminous data through conventional centralized system is bit complex. It means, even simple queries are time consuming queries. The solution is to handle those databases through Parallel Database Systems, where a table / database is distributed among multiple processors possibly equally to perform the queries in parallel. Such a system which share resources to handle massive data just to increase the performance of the whole system is called Parallel Database Systems.

We need certain architecture to handle the above said. That is, we need architectures which can handle data through data distribution, parallel query execution thereby produce good throughput of queries or Transactions. Figure 1, 2 and 3 shows the different architecture proposed and successfully implemented in the area of Parallel Database systems. In the figures, P represents Processors, M represents Memory, and D represents Disks/Disk setups.

1. Shared Memory Architecture

Figure 1 - Shared Memory Architecture



In Shared Memory architecture, single memory is shared among many processors as show in Figure 1. As shown in the figure, several processors are connected through an interconnection network with Main memory and disk setup. Here interconnection network is usually a high speed network (may be Bus, Mesh, or Hypercube) which makes data sharing (transporting) easy among the various components (Processor, Memory, and Disk).

Advantages:

- Simple implementation
- Establishes effective communication between processors through single memory addresses space.
- Above point leads to less communication overhead.

Disadvantages:

- Higher degree of parallelism (more number of concurrent operations in different processors) cannot be achieved due to the reason that all the processors share the same interconnection network to connect with memory. This causes Bottleneck in interconnection network (Interference), especially in the case of Bus interconnection network.
- Addition of processor would slow down the existing processors.
- Cache-coherency should be maintained. That is, if any processor tries to read the data used or modified by other processors, then we need to ensure that the data is of latest version.
- Degree of Parallelism is limited. More number of parallel processes might degrade the performance.

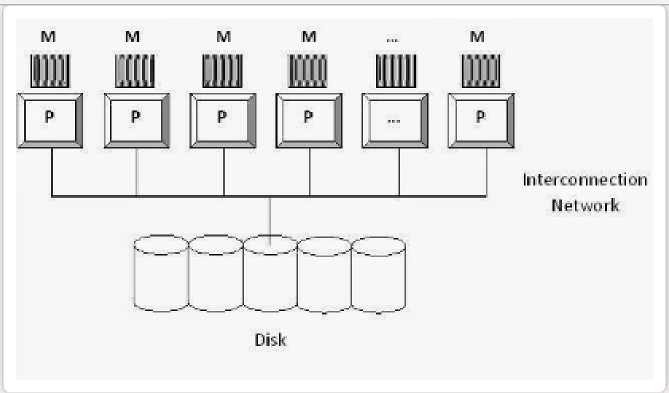


Figure 2 - Shared Disk Architecture

In Shared Disk architecture, single disk or single disk setup is shared among all the available processors and also all the processors have their own private memories as shown in Figure 2.

Advantages:

- Failure of any processors would not stop the entire system (Fault tolerance)
- Interconnection to the memory is not a bottleneck. (It was bottleneck in Shared Memory architecture)
- Support larger number of processors (when compared to Shared Memory architecture)

Disadvantages:

- Interconnection to the disk is bottleneck as all processors share common disk setup.
- Inter-processor communication is slow. The reason is, all the processors have their own memory. Hence, the communication between processors need reading of data from other processors' memory which needs additional software support.

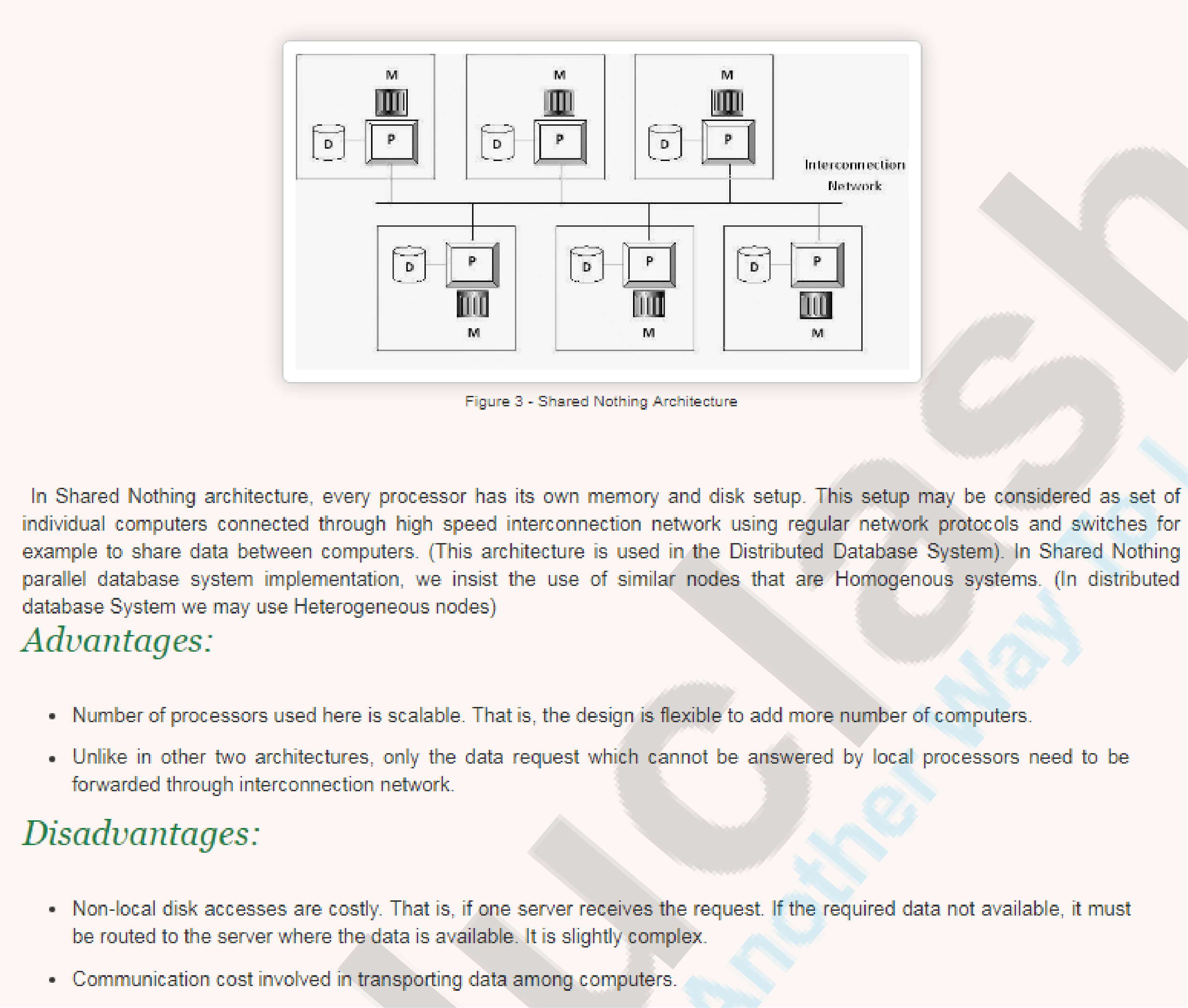


Figure 3 - Shared Nothing Architecture

In Shared Nothing architecture, every processor has its own memory and disk setup. This setup may be considered as set of individual computers connected through high speed interconnection network using regular network protocols and switches for example to share data between computers. (This architecture is used in the Distributed Database System). In Shared Nothing parallel database system implementation, we insist the use of similar nodes that are Homogenous systems. (In distributed database System we may use Heterogeneous nodes)

Advantages:

- Number of processors used here is scalable. That is, the design is flexible to add more number of computers.
- Unlike in other two architectures, only the data request which cannot be answered by local processors need to be forwarded through interconnection network.

Disadvantages:

- Non-local disk accesses are costly. That is, if one server receives the request. If the required data not available, it must be routed to the server where the data is available. It is slightly complex.
- Communication cost involved in transporting data among computers.

Techniques of query Evaluation

The two techniques used in query evaluation are as follows:

1. Inter query parallelism

- This technique allows to run multiple queries on different processors simultaneously.
- Pipelined parallelism is achieved by using inter query parallelism, which improves the output of the system.

For example: If there are 6 queries, each query will take 3 seconds for hevaluation. Thus, the total time taken to complete evaluation process is 18 seconds. Inter query parallelism achieves this task only in 3 seconds.

- However, Inter query parallelism is difficult to achieve every time.

2. Intra Query Parallelism

- In this technique query is divided in sub queries which can run simultaneously on different processors, this will minimize the query evaluation time.
- Intra query parallelism improves the response time of the system.

For Example: If we have 6 queries, which can take 3 seconds to complete the evaluation process, the total time to complete the evaluation process is 18 seconds. But We can achieve this task in only 3 seconds by using intra query evaluation as each query is divided in sub-queries.

Optimization of Parallel Query

- Parallel Query optimization is nothing but selecting the efficient query evaluation plan.
- Parallel Query optimization plays an important role in developing system to minimize the cost of query evaluation.

Two factors play a very important in parallel query optimization.

a) total time spent to find the best plan.

b) amount of time required to execute the plan.

Goals of Query optimization.

Query Optimization is done with an aim to:

- Speed up the queries by finding the queries which can give the fastest result on execution.
- Increase the performance of the system.
- Select the best query evaluation plan.
- Avoid the unwanted plan.

Approaches of Query Optimization.

Following are the three approaches to Query Optimization:

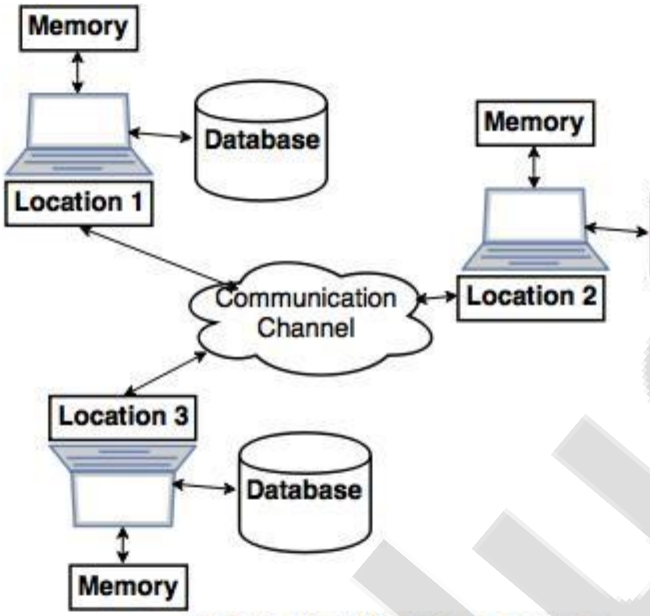
1. Horizontal partitioning: Tables are created vertically using columns.

2. Vertical partitioning: Tables are created with fewer columns and partition the table row wise.

3. De-normalization: In this approach multiple tables are combined into one table.

What are distributed databases?

- Distributed database is a system in which storage devices are not connected to a common processing unit.
- Database is controlled by Distributed Database Management System and data may be stored at the same location or spread over the interconnected network. It is a loosely coupled system.
- Shared nothing architecture is used in distributed databases.



Distributted Dtatabase system

- The above diagram is a typical example of distributed database system, in which communication channel is used to communicate with the different locations and every system has its own memory and database.

Goals of Distributed Database system.

The concept of distributed database was built with a goal to improve:

Reliability: In distributed database system, if one system fails down or stops working for some time another system can complete the task.
Availability: In distributed database system reliability can be achieved even if sever fails down. Another system is available to serve the client request.
Performance: Performance can be achieved by distributing database over

different locations. So the databases are available to every location which is easy to maintain.

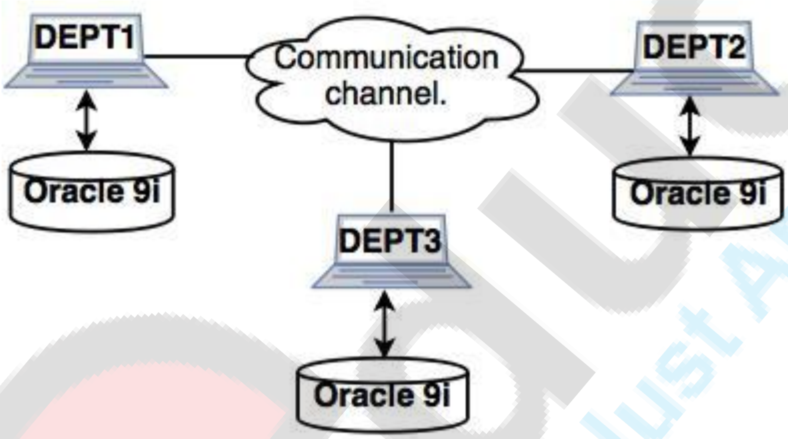
Types of distributed databases.

The two types of distributed systems are as follows:

1. Homogeneous distributed databases system:

- Homogeneous distributed database system is a network of two or more databases (With same type of DBMS software) which can be stored on one or more machines.
- So, in this system data can be accessed and modified simultaneously on several databases in the network. Homogeneous distributed system are easy to handle.

Example: Consider that we have three departments using Oracle-9i for DBMS. If some changes are made in one department then, it would update the other department also.

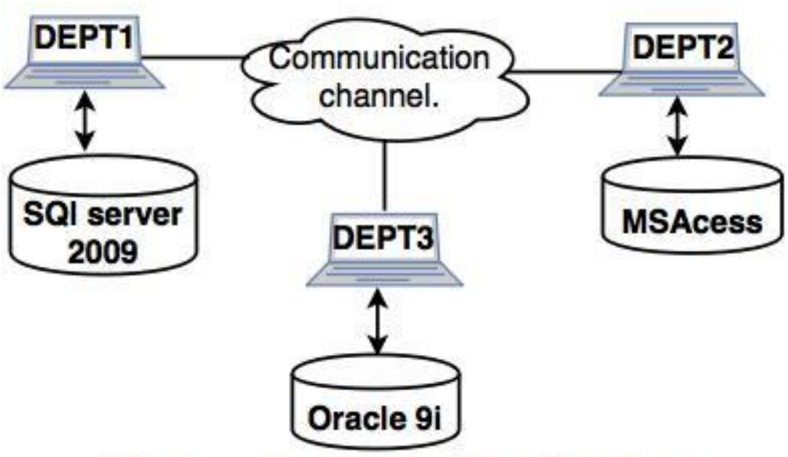


Homogeneous distributed system

2. Heterogeneous distributed database system.

- Heterogeneous distributed database system is a network of two or more databases with different types of DBMS software, which can be stored on one or more machines.
- In this system data can be accessible to several databases in the network with the help of generic connectivity (ODBC and JDBC).

Example: In the following diagram, different DBMS software are accessible to each other using ODBC and JDBC.



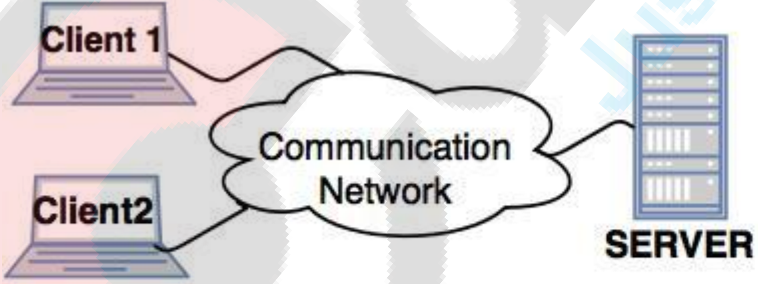
Heterogeneous distributed system

Architectures of Distributed DBMS

The basic types of distributed DBMS are as follows:

1. Client-server architecture of Distributed system.

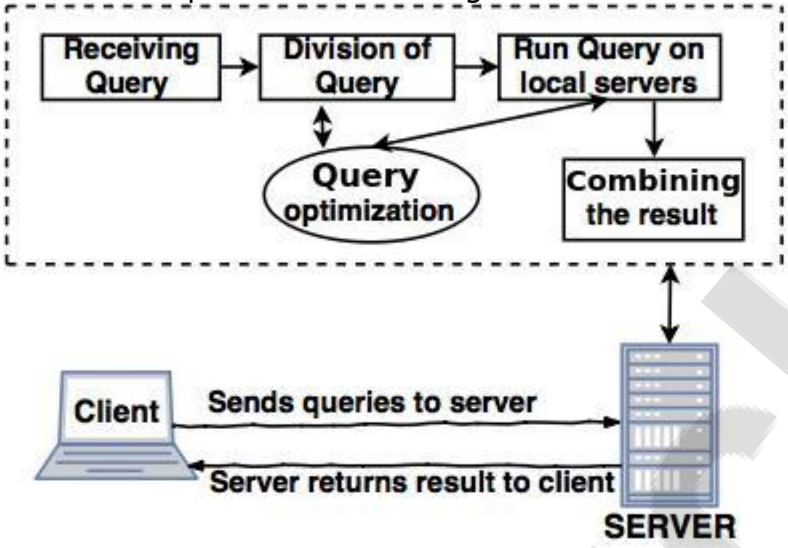
- A client server architecture has a number of clients and a few servers connected in a network.
- A client sends a query to one of the servers. The earliest available server solves it and replies.
- A Client-server architecture is simple to implement and execute due to centralized server system.



Client-server architecture

2. Collaborating server architecture.

- Collaborating server architecture is designed to run a single query on multiple servers.
- Servers break single query into multiple small queries and the result is sent to the client.
- Collaborating server architecture has a collection of database servers. Each server is capable for executing the current transactions across the databases.



Collabrorating server architecture

3. Middleware architecture.

- Middleware architectures are designed in such a way that single query is executed on multiple servers.
- This system needs only one server which is capable of managing queries and transactions from multiple servers.
- Middleware architecture uses local servers to handle local queries and transactions.
- The softwares are used for execution of queries and transactions across one or more independent database servers, this type of software is called as middleware.

What is recovery in distributed databases?

Recovery is the most complicated process in distributed databases. Recovery of a failed system in the communication network is very difficult.

For example:
Consider that, location A sends message to location B and expects response from B but B is unable to receive it. There are several problems for this situation which are as follows.

- Message was failed due to failure in the network.
- Location B sent message but not delivered to location A.
- Location B crashed down.
- So it is actually very difficult to find the cause of failure in a large communication network.
- Distributed commit in the network is also a serious problem which can affect the recovery in a distributed databases.

Two-phase commit protocol in Distributed databases

- Two-phase protocol is a type of atomic commitment protocol. This is a distributed algorithm which can coordinate all the processes that participate in the database and decide to commit or terminate the transactions. The protocol is based on commit and terminate action.
- The two-phase protocol ensures that all participant which are accessing the database server can receive and implement the same action (Commit or terminate), in case of local network failure.
- Two-phase commit protocol provides automatic recovery mechanism in case of a system failure.
- The location at which original transaction takes place is called as coordinator and where the sub process takes place is called as **Cohort.**

Commit request:
In commit phase the coordinator attempts to prepare all cohorts and take necessary steps to commit or terminate the transactions.

Commit phase:

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

Concurrency problems in distributed databases.

Some problems which occur while accessing the database are as follows:

1. Failure at local locations
When system recovers from failure the database is out dated compared to other locations. So it is necessary to update the database.

2. Failure at communication location
System should have a ability to manage temporary failure in a communicating network in distributed databases. In this case, partition occurs which can limit the communication between two locations.

3. Dealing with multiple copies of data
It is very important to maintain multiple copies of distributed data at different locations.

4. Distributed commit
While committing a transaction which is accessing databases stored on multiple locations, if failure occurs on some location during the commit process then this problem is called as distributed commit.

5. Distributed deadlock
Deadlock can occur at several locations due to recovery problem and concurrency problem (multiple locations are accessing same system in the communication network).

Concurrency Controls in distributed databases

There are three different ways of making distinguish copy of data by applying:

1) Lock based protocol
A lock is applied to avoid concurrency problem between two transaction 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 or read operations. It is an important method to avoid deadlock.

2) Shared lock system (Read lock)

The transaction can activate shared lock on 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
The transaction can activate exclusive lock on a data to read and write operation. In this system, no other transaction can activate any kind of lock on that same data.

Distributed Transactions

- A Distributed Databases Management System should be able to survive in a system failure without losing any data in the database.
- This property is provided in transaction processing.
- The local transaction works only on own location(Local Location) where it is considered as a global transaction for other locations.
- Transactions are assigned to transaction monitor which works as a supervisor.
- A distributed transaction process is designed to distribute data over many locations and transactions are carried out successfully or terminated successfully.
- Transaction Processing is very useful for concurrent execution and recovery of data.

What is data replication?

Data replication is the process in which the data is copied at multiple locations (Different computers or servers) to improve the availability of data.

Goals of data replication

Data replication is done with an aim to:

- Increase the availability of data.
- Speed up the query evaluation.

Types of data replication

There are two types of data replication:

1. Synchronous Replication:
In synchronous replication, the replica will be modified immediately after some changes are made in the relation table. So there is no difference between original data and replica.

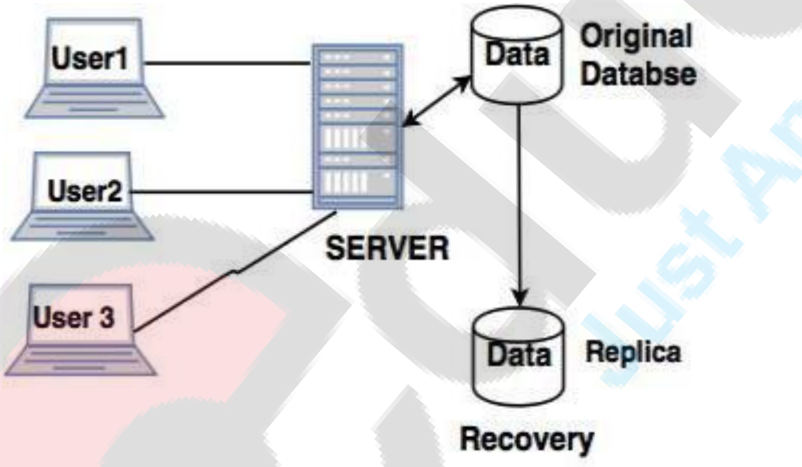
2. Asynchronous replication:
In asynchronous replication, the replica will be modified after commit is fired on to the database.

Replication Schemes

The three replication schemes are as follows:

1. Full Replication

In full replication scheme, the database is available to almost every location or user in communication network.



Full Replication Process In Distributed System

Advantages of full replication

- High availability of data, as database is available to almost every location.
- Faster execution of queries.

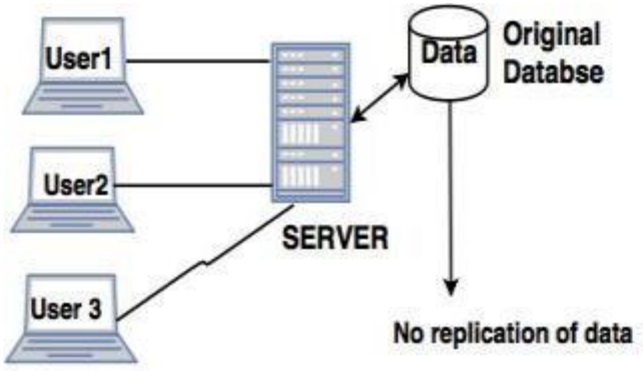
Disadvantages of full replication

- Concurrency control is difficult to achieve in full replication.

- Update operation is slower.

2. No Replication

No replication means, each fragment is stored exactly at one location.



No Replication Process in Distributed Databases

Advantages of no replication

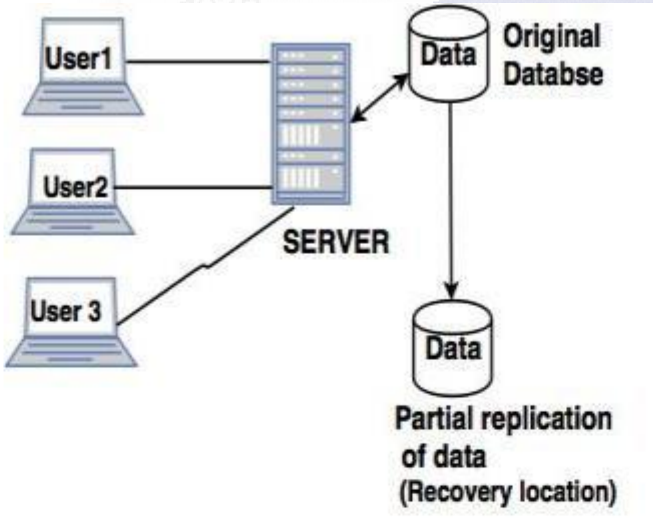
- Concurrency can be minimized.
- Easy recovery of data.

Disadvantages of no replication

- Poor availability of data.
- Slows down the query execution process, as multiple clients are accessing the same server.

3. Partial replication

Partial replication means only some fragments are replicated from the database.



Partial Replication Process In Distributed System

Advantages of partial replication
The number of replicas created for fragments depend upon the importance of data in that fragment.

Distributed databases - Query processing and Optimization

DDBMS processes and optimizes a query in terms of communication cost of processing a distributed query and other parameters.

Various factors which are considered while processing a query are as follows:

Costs of Data transfer

- This is a very important factor while processing queries. The intermediate data is transferred to other location for data processing and the final result will be sent to the location where the actual query is processing.
- The cost of data increases if the locations are connected via high performance communicating channel.
- The DDBMS query optimization algorithms are used to minimize the cost of data transfer.

Semi-join based query optimization

- Semi-join is used to reduce the number of relations in a table before transferring it to another location.
- Only joining columns are transferred in this method.
- This method reduces the cost of data transfer.

Cost based query optimization

- Query optimization involves many operations like, selection, projection, aggregation.
- Cost of communication is considered in query optimization.
- In centralized database system, the information of relations at remote location is obtained from the server system catalogs.
- The data (query) which is manipulated at local location is considered as a sub query to other global locations. This process estimates the total cost which is needed to compute the intermediate relations.