REPLICATION

Replication is the process of storing data in more than one site or node. It is useful in improving the availability of data. It is simply copying data from a database from one server to another server so that all the users can share the same data without any inconsistency. The result is a distributed database in which users can access data relevant to their tasks without interfering with the work of others.

Reasons for Data Replication

- Availability
  - at least some server somewhere
  - wireless connections => a local cache
- reliability (correctness of data)
  - fault tolerance against data corruption
  - fault tolerance against faulty operations
- Better performance
  1. Multiple servers offer the same service – parallel processing of client requests
- Geographical distribution
  1. Creating copies of data/objects closer to the clients leads to smaller network delay and possibly reduced network traffic

There are two types of data replication:

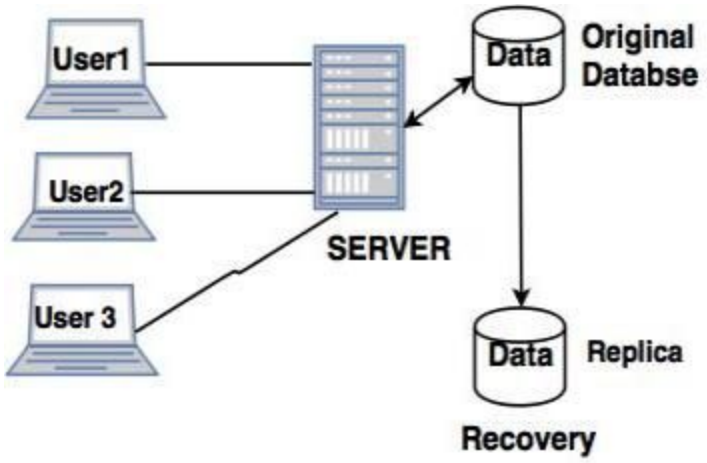
1. Synchronous Replication:
In synchronous replication, the replica will be modified immediately after some changes are made in the relation table. So there is no difference between original data and replica.

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

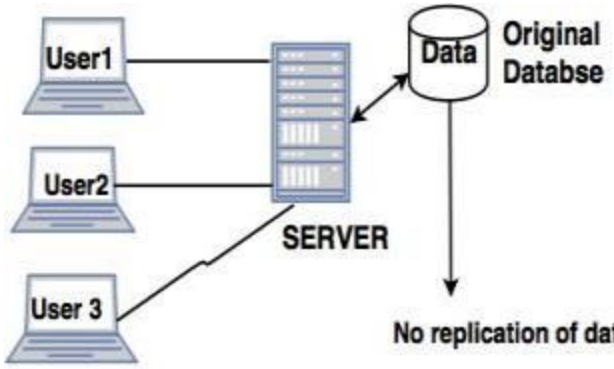
Advantages of full replication

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

Disadvantages of full replication

1. Concurrency control is difficult to achieve in full replication.
2. 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

1. Concurrency can be minimized.
2. Easy recovery of data.

Disadvantages of no replication

1. Poor availability of data.
2. 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.

Advantages of partial replication

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

ADVANTAGES OF DATA REPLICATION

1. To provide a consistent copy of data across all the database nodes.
2. To increase the availability of data.
3. The reliability of data is increased through data replication.
4. Data Replication supports multiple users and gives high performance.
5. To remove any data redundancy,the databases are merged and slave databases are updated with outdated or incomplete data.
6. Since replicas are created there are chances that the data is found itself where the transaction is executing which reduces the data movement.
7. To perform faster execution of queries.

DISADVANTAGES OF DATA REPLICATION –

1. More storage space is needed as storing the replicas of same data at different sites consumes more space.
2. Data Replication becomes expensive when the replicas at all different sites need to be updated.
3. Maintaining Data consistency at all different sites involves complex measures.

The **replica manager** is a subsystem that is responsible for managing the synchronization of replicas. The replica manager is responsible for the following:

1. Marshaling and unmarshaling the replicas in each peer.
2. Forwarding replicas from one peer to another.
3. Handling ownership changes of replicas
4. Managing replica lifetimes

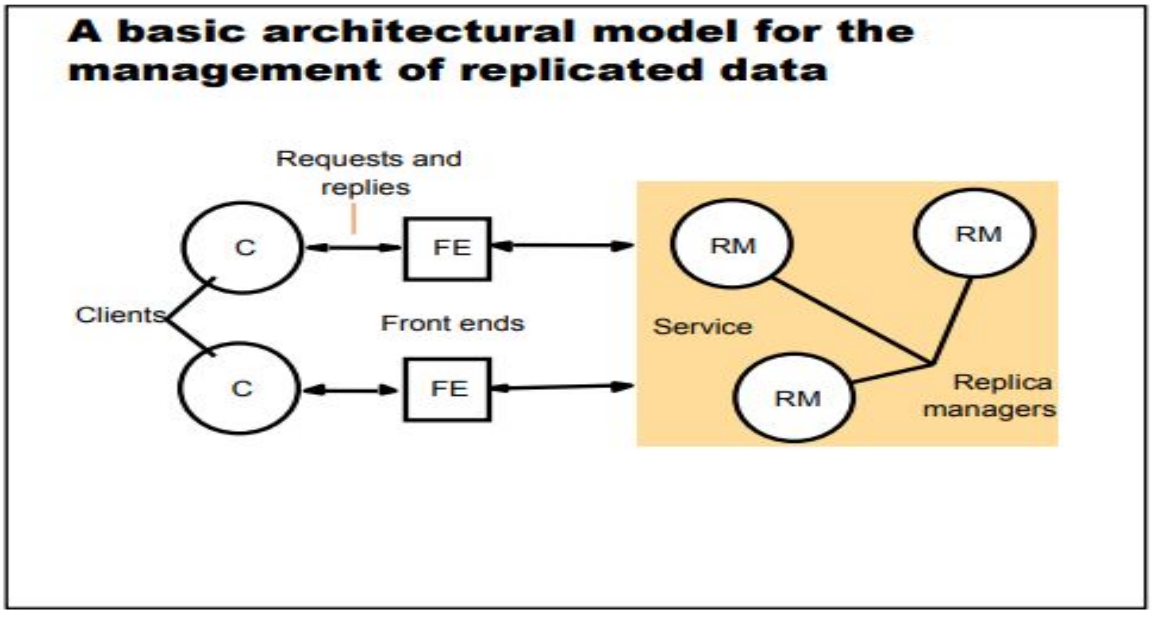
Active replication, which is performed by processing the same request at every replica
Passive replication, which involves processing every request on a single replica and transferring the result to the other replicas

Three widely cited models exist for data replication, each having its own properties and performance:

Transactional replication: used for replicating transactional data, such as a database. The one-copy serializability model is employed, which defines valid outcomes of a transaction on replicated data in accordance with the overall ACID (atomicity, consistency, isolation, durability) properties that transactional systems seek to guarantee.

State machine replication: assumes that the replicated process is a deterministic finite automaton and that atomic broadcast of every event is possible. It is based on distributed consensus and has a great deal in common with the transactional replication model. This is sometimes mistakenly used as a synonym of active replication. State machine replication is usually implemented by a replicated log consisting of multiple subsequent rounds of the Paxos algorithm. This was popularized by Google's Chubby system, and is the core behind the open-source Keyspace data store.

Virtual synchrony: involves a group of processes which cooperate to replicate in-memory data or to coordinate actions. The model defines a distributed entity called a process group. A process can join a group and is provided with a checkpoint containing the current state of the data replicated by group members. Processes can then send multicasts to the group and will see incoming multicasts in the identical order. Membership changes are handled as a special multicast that delivers a new "membership view" to the processes in the group.



Types of Data Replication –

Transactional Replication – In Transactional replication users receive full initial copies of the database and then receive updates as data changes. Data is copied in real time from the publisher to the receiving database(subscriber) in the same order as they occur with the publisher therefore in this type of replication, transactional consistency is guaranteed. Transactional replication is typically used in server-to-server environments. It does not simply copy the data changes, but rather consistently and accurately replicates each change.

Snapshot Replication – Snapshot replication distributes data exactly as it appears at a specific moment in time does not monitor for updates to the data. The entire snapshot is generated and sent to Users. Snapshot replication is generally used when data changes are infrequent. It is bit slower than transactional because on each attempt it moves multiple records from one end to the other end. Snapshot replication is a good way to perform initial synchronization between the publisher and the subscriber.

Merge Replication – Data from two or more databases is combined into a single database. Merge replication is the most complex type of replication because it allows both publisher and subscriber to independently make changes to the database. Merge replication is typically used in server-to-client environments. It allows changes to be sent from one publisher to multiple subscribers.

Fault Tolerance Techniques Replication • Creating multiple copies or replica of data items and storing them at different sites • Main idea is to increase the availability so that if a node fails at one site, so data can be accessed from a different site