

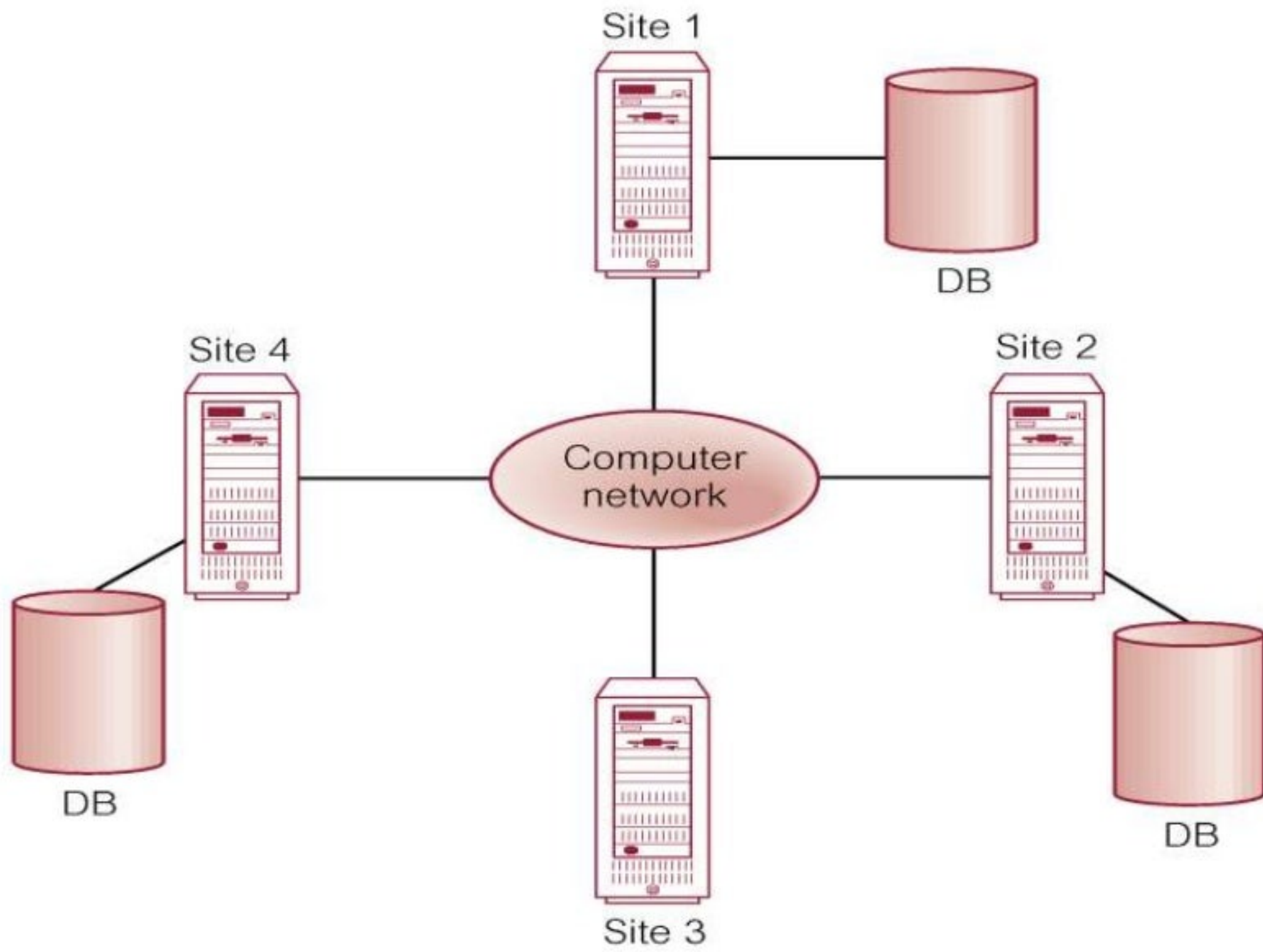



Distributed Database

- Database:- Logical interrelated collection of shared data, along with description of data, physically distributed over a computer network.

What is Distributed Database?

- A distributed database (DDB) is a collection of multiple, logically interrelated databases distributed over a computer network.
- A distributed database management system (DDBMS) is the software that manages the DDB and provides an access mechanism that makes this distribution transparent to the users



- 
- A DDBMS mainly classified into two types:
 - Homogeneous Distributed database management systems
 - Heterogeneous Distributed database management systems



Characteristics

- All sites are interconnected.
- Fragments can be replicated.
- Logically related shared data can be collected.
- Data at each and every site is controlled by the DBMS.
- Each Distributed Database Management System takes part in at least one global application.

Functionality

- Security
- Keeping track of data
- Replicated data management
- System catalog management
- Distributed transaction management
- Distributed database recovery

Homogeneous DDBMS

- In a homogeneous distributed database all sites have identical software and are aware of each other and agree to cooperate in processing user requests.
- The homogeneous system is much easier to design and manage
- The operating system used, at each location must be same or compatible.
- The database application (or DBMS) used at each location must be same or compatible.

Heterogeneous DDBMS

- In a heterogeneous distributed database different sites may use different schema and software.
- In heterogeneous systems, different nodes may have different hardware & software and data structures at various nodes or locations are also incompatible.
- Different computers and operating systems, database applications or data models may be used at each of the locations.

Heterogeneous DDBMS (contd..)

- On heterogeneous system, translations are required to allow communication between different sites (or DBMS).
- The heterogeneous system is often not technically or economically feasible. In this system, a user at one location may be able to read but not update the data at another location.

Advantages

- Less danger of a single-point failure. When one of the computers fails, the workload is picked up by other workstations.
- Data are also distributed at multiple sites.
- The end user is able to access any available copy of the data, and an end user's request is processed by any processor at the data location.

Advantages (contd..)

- Improved communications. Because local sites are smaller and located closer to customers.
- Reduced operating costs. It is more cost-effective to add workstations to a network than to update a mainframe system.
- Faster data access, faster data processing.
- A distributed database system spreads out the systems workload by processing data at several sites.



Disadvantages

- Complexity of management and control.
- Applications must recognize data location, and they must be able to stitch together data from various sites.
- Security.

Disadvantages (contd..)

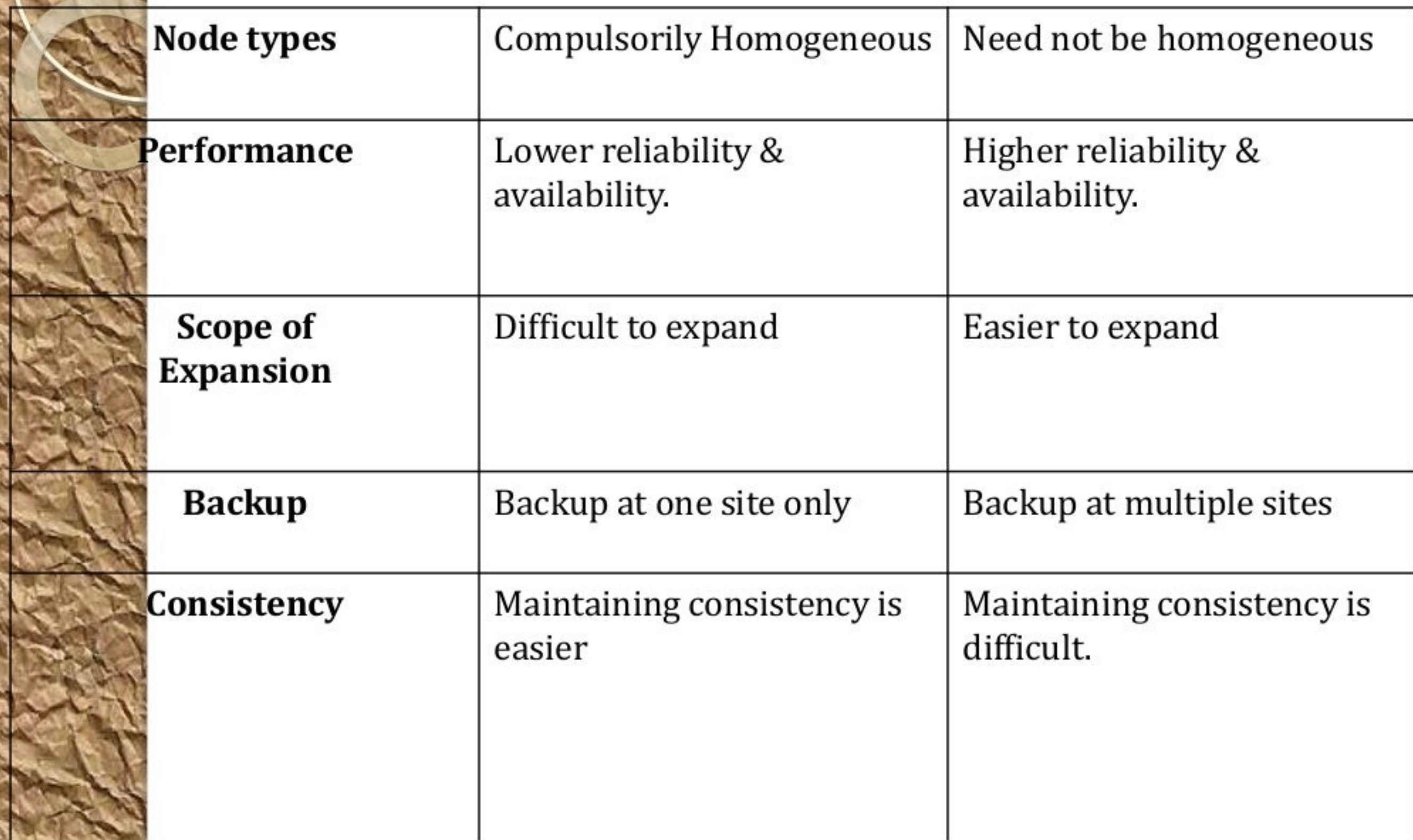
- Increased storage and infrastructure requirements.
- Multiple copies of data has to be at different sites, thus an additional disk storage space will be required.
- The probability of security lapses increases when data are located at multiple sites.

What is Parallel database...??

- A parallel database system is to improve performance through parallelization of various operations, such as loading data, building indexes and evaluating queries.
- The distribution is solely done on the bases of performance.
- Parallel databases improve processing and input/output speeds by using multiple CPUs and disks in parallel.
- Many operations are performed simultaneously
- Data may be stored in a distributed fashion.

Difference b/w Distributed Database and Parallel Database

Characteristics	Parallel Database	Distributed database
Definition	It is a software system where multiple processors or machines are used to execute and run queries in parallel.	It is a software system that manages multiple logically interrelated databases distributed over a computer network.
Geographical Location	The nodes are located at geographically same location.	The nodes are usually located at geographically different locations.
Execution Speed	Quicker	Slower
Overhead	Less	More



Node types	Compulsorily Homogeneous	Need not be homogeneous
Performance	Lower reliability & availability.	Higher reliability & availability.
Scope of Expansion	Difficult to expand	Easier to expand
Backup	Backup at one site only	Backup at multiple sites
Consistency	Maintaining consistency is easier	Maintaining consistency is difficult.

Data fragmentation

- Fragmentation is a process of division or the mapping of the tables based on the columns and rows of data into the smallest unit of data.
- Data that has broken down is still possible to be combined again with the intention to complete the data collection using fragmentation.
- Fragmentation is a database server feature that allows you to control where data is stored at the table level.
- Fragmentation enables you to define groups of rows or index keys within a table.

Replication

- Replication is that we store several copies of a relation or relation fragment. An entire relation can be replicated at one or more sites.
- Similarly, one or more fragments of a relation can be replicated at other sites.
- For example, if a relation R is fragmented into R_1, R_2 , and R_3 , *there might be just one copy of R_1 , whereas R_2 is replicated at two other sites and R_3 is replicated at all sites.*

Two Fold Replication


The motivation for replication is twofold:

1. **Increased Availability of Data:** If a site that contains a replica goes down, we can find the same data at other sites. Similarly, if local copies of remote relations are available, we are less vulnerable to failure of communication links.
2. **Faster Query Evaluation:** Queries can execute faster by using a local copy of a relation instead of going to a remote site.




Distributed Transaction

- In a distributed DBMS, a given transaction is submitted at some one site, but it can access data at other sites as well.
- When a transaction is submitted at some site, the transaction manager at that site breaks it up into a collection of one or more sub-transactions that execute at different sites, submits them to transaction managers at the other sites, and coordinates their activity.

- 
- Distributed Concurrency Control: How can locks for objects stored across several sites be managed?
 - Distributed Recovery: Transaction atomicity must be ensured when a transaction commits, all its actions, across all the sites at which it executes, must persist. Similarly, when a transaction aborts, none of its actions must be allowed to persist.

Distributed Concurrency Control

- The choice of technique determines *which* objects are to be locked. *When locks are obtained and released is determined* by the concurrency control protocol. We now consider how lock and unlock requests are implemented in a distributed environment. Lock management can be distributed across sites in many ways:

- 
- Centralized : A single site is in charge of handling lock and unlock requests for all objects.
 - Primary Copy: One copy of each object is designated the primary copy. All requests to lock or unlock a copy of this object are handled by the lock manager at the site where the primary copy is stored, regardless of where the copy itself is stored.
 - Fully Distributed : Requests to lock or unlock a copy of an object stored at a site are handled by the lock manager at the site where the copy is stored.

DISTRIBUTED RECOVERY

- Recovery in a distributed DBMS is more complicated than in a centralized DBMS for the following reasons:
 - New kinds of failure can arise : Failure of communication links and failure of a remote site at which a sub-transaction is executing.
 - Either all sub-transactions of a given transaction must commit or none must commit, and this property must be guaranteed despite any combination of site and link failures. This guarantee is achieved using a commit protocol.

Concepts Of Locks

- A **lock** is used when multiple users need to access a database concurrently. This prevents data from being corrupted or invalidated when multiple users try to write to the database.
- Any single user can only modify those database records (that is, items in the database) to which they have applied a lock that gives them exclusive access to the record until the lock is released. Locking not only provides exclusivity to write but also prevents (or controls) reading of unfinished modifications.

Thank You.

