UNIT–V ADVANCED TOPICS

Distributed Databases: Architecture(L2) – Types of Distributed Databases(L2) – Transaction Processing(L2). NoSQL Databases: Introduction(L2) – CAP Theorem(L2) – Document Based Systems(L2) – Key Value Stores(L2) – Column Based Systems(L2) – Graph Databases(L2).Database Security: Security Issues(L2) – Access Control Based on Privileges(L2) – Role Based Access Control(L2) – SQL Injection(L2) – Encryption and Public Key Infrastructures(L2) – Challenges(L2).

DISTRIBUTED DATABASES

Definition of distributed databases:

• A distributed database system consists of loosely coupled sites (computer) that share no physical components and each site is associated a database system.

• The software that maintains and manages the working of distributed databases is called distributed database management system.

• The database system that runs on each site is independent of each other. Refer Fig. 5.1.1.

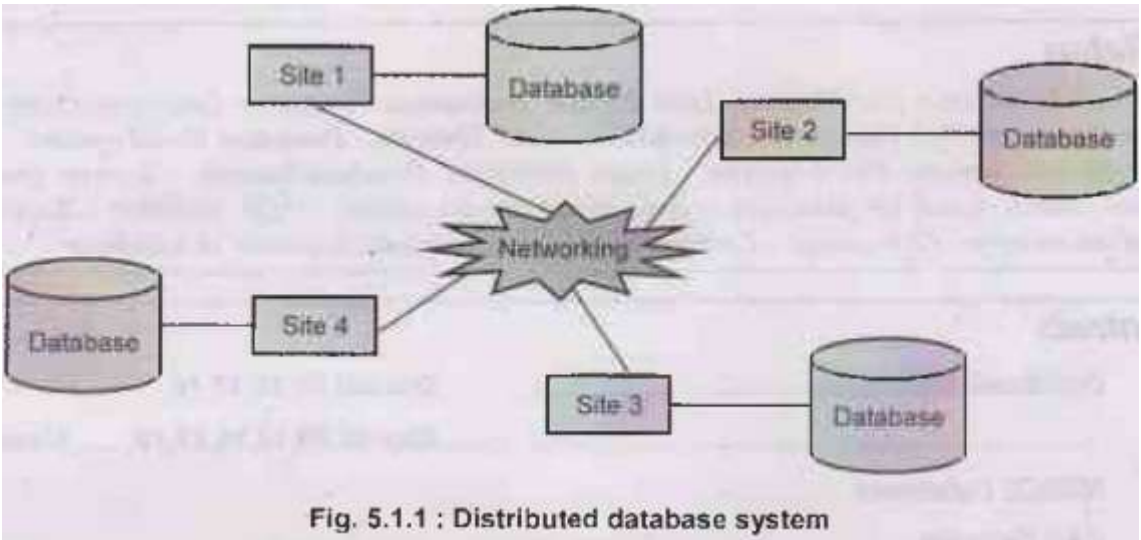


Fig. 5.1.1 : Distributed database system

The transactions can access data at one or more sites.

Advantages of distributed database system

(1) There is fast data processing as several sites participate in request processing. 2.

(2) Reliability and availability of this system is high.

(3) It possess reduced operating cost.

(4) It is easier to expand the system by adding more sites.

(5) It has improved sharing ability and local autonomy.

Disadvantages of distributed database system

(1) The system becomes complex to manage and control.

(2) The security issues must be carefully managed.

(3) The system require deadlock handling during the transaction processing otherwise the entire system may be in inconsistent state.

(4) There is need of some standardization for processing of distributed database system.

Difference between distributed DBMS and centralized DBMS

Sr. No.	Distributed DBMS	Centralized DBMS
1.	The database files are stored at geographically different locations across the network.	The database is stored at centralized location.
2.	As data is distributed over the network , it requires time to synchronize data and thus difficult to maintain.	A centralized database is easier to maintain and keep updated since all the data are stored in a single location.
3.	If one database fails, user can have access to other database files.	If the centralized database fails, then there is no access to a database.
4.	It can have data replication as database is distributed. Hence there can be some data inconsistency.	It have single database system, hence there is no data replication. Therefore there is no data inconsistency.

Uses of distributed system:

(1) Often distributed databases are used by organizations that have numerous offices in different geographical locations. Typically an individual branch is interacting primarily with the data that pertain to its own operations, with a much less frequent need for general company data. In such a situation, distributed systems are useful.

(2) Using distributed system, one can give permissions to single sections of the overall database, for better internal and external protection.

(3) If we need to add a new location to a business, it is simple to create an additional node within the database, making distribution highly scalable.

Types of Distributed Databases

There are two types of distributed databases -

(1) Homogeneous databases

• The homogeneous databases are kind of database systems in which all sites have identical software running on them. Refer Fig. 5.1.2.

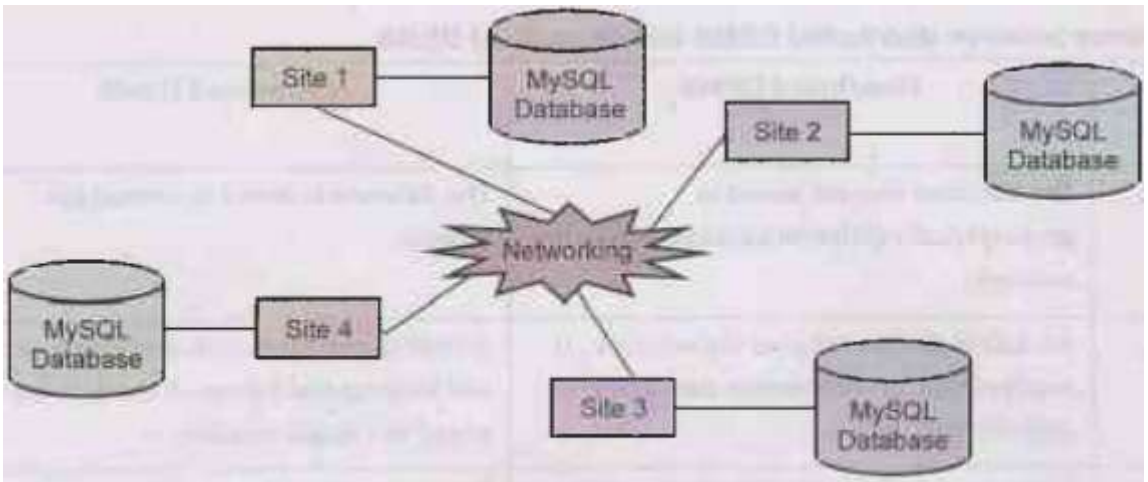


Fig. 5.1.2 : Homogeneous databases

• In this system, all the sites are aware of the other sites present in the system and they all cooperate in processing user's request.

• Each site present in the system, surrenders part of its autonomy in terms of right to change schemas or software.

• The homogeneous database system appears as a single system to the user.

(2) Heterogeneous databases

• The heterogeneous databases are kind of database systems in which different sites have different schema or software. Refer Fig. 5.1.3.

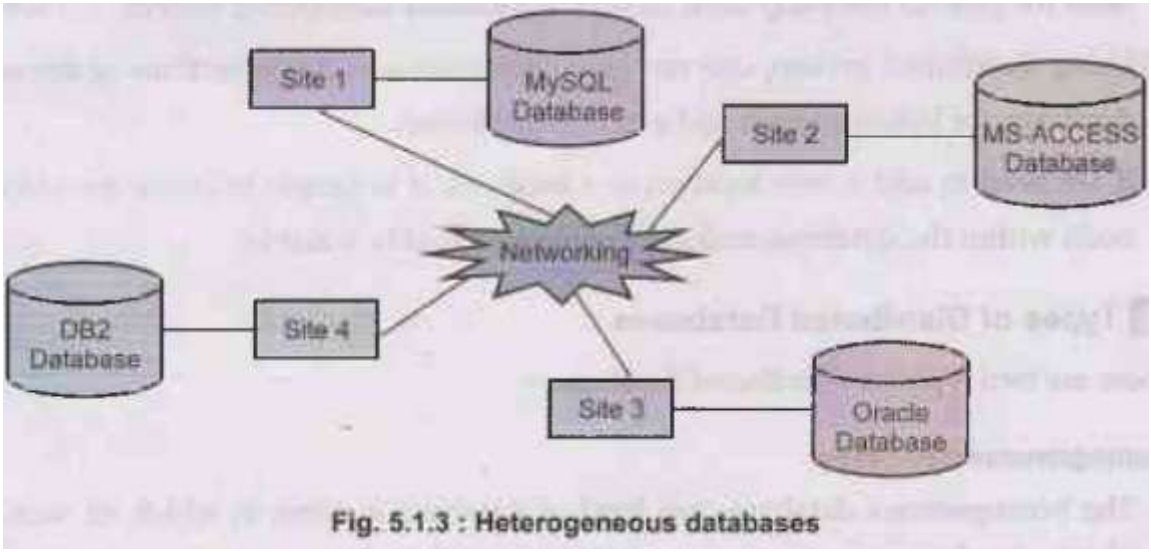


Fig. 5.1.3 : Heterogeneous databases

• The participating sites are not aware of other sites present in the system.

• These sites provide limited facilities for cooperation in transaction processing.

Architecture

• Following is an architecture of distributed databases. In this architecture the local database is maintained by each site.

• Each site is interconnected by communication network.

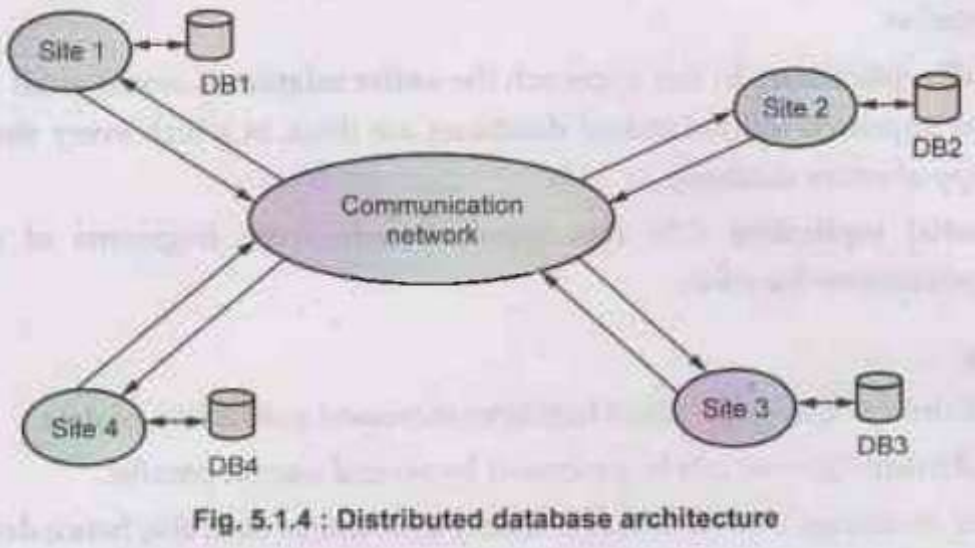


Fig. 5.1.4 : Distributed database architecture

When user makes a request for particular data at site Si then it is first searched at the local database. If the data is not present in the local database then the request for that data is passed to all the other sites via communication network. Each site then searches for that data at its local database. When data is found at particular site say Sj then it is transmitted to site Si via communication network.

TRANSACTION PROCESSING

Basic Concepts

In distributed system transaction initiated at one site can access or update data at other sites. Let us discuss various basic concepts used during transaction processing in distributed systems -

• **Local and global transactions :**

Local transaction Ti is said to be local if it is initiated at site Si and can access or update data at site Si only.

Global transaction Ti initiated by site Si is said to be global if it can access or update data at site Si, Sj,Sk and so on.

• **Coordinating and participating sites:**

The site at which the transaction is initiated is called coordinating site. The participating sites are those sites at which the sub-transactions are executing. For example - If site S1 initiates the transaction T1 then it is called coordinating site. Now assume that transaction T1 (initiated at S1) can access site S2 and S3. Then sites S2 and S3 are called participating sites.

To access the data on site S2, the transaction T1 needs another transaction T12 on site S2 similarly to access the data on site S3, the transaction T2 needs some transaction say T13 on site S3. Then transactions T12 and T13 are called sub-

transactions. The above described scenario can be represented by following Fig. 5.1.6.

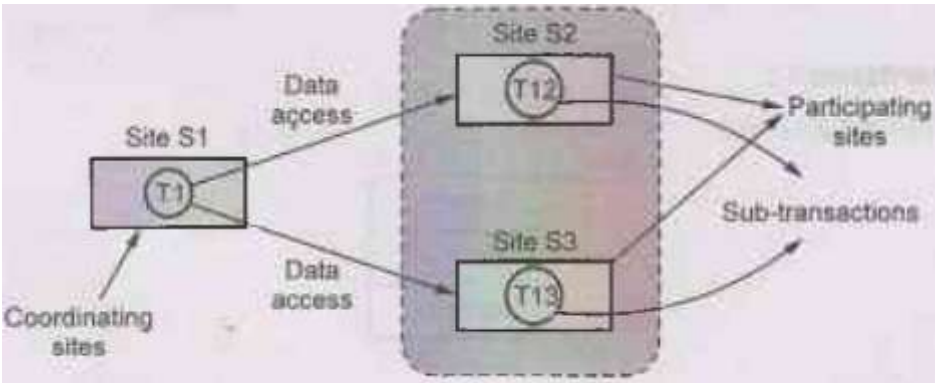


Fig. 5.1.6 : Distributed transaction scenario

• **Transaction manager :**

The transaction manager manages the execution of those transactions (or subtransactions) that access data stored in a local site.

(1) To maintain the log for recovery purpose.

(2) Participating in coordinating the concurrent execution of the transactions executing balls at that site.

• **Transaction coordinator:**

The transaction coordinator coordinates the execution of the various transactions (both local and global) initiated at that site.

The tasks of Transaction coordinator are -

(1) Starting the execution of transactions that originate at the site.

(2) Distributing subtransactions at appropriate sites for execution

Let TC denotes the transaction coordinator and TM denotes the transaction manager, then the system architecture can be represented as,

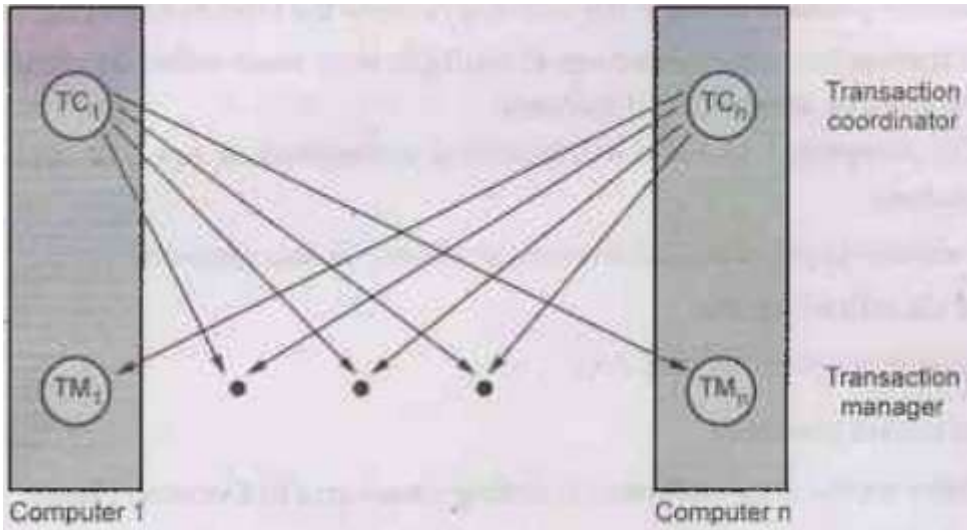


Fig. 5.1.7 : System architecture

Failure Modes

There are four types of failure modes,

1. Failure of site

2. Loss of messages

3. Failure of communication link

4. Network partition

The most common type of failure in distributed system is loss or corruption of messages. The system uses Transmission Control Protocol(TCP) to handle such error. This is a standard connection oriented protocol in which message is transmitted from one end to another using wired connection.

• If two nodes are not directly connected, messages from one to another must be routed through sequence of communication links. If the communication link fails, the messages are rerouted by alternative links.

• A system is partitioned if it has been split into two subsystems. This is called partitions. Lack of connection between the subsystems also cause failure in distributed system.

Commit Protocols

Two Phase Commit Protocol

• The atomicity is an important property of any transaction processing. What is this atomicity property? This property means either the transaction will execute completely or it won't execute at all.

• The commit protocol ensures the atomicity across the sites in following ways -

i) A transaction which executes at multiple sites must either be committed at all the sites, or aborted at all the sites.

ii) Not acceptable to have a transaction committed at one site and aborted at another.

• There are two types of important sites involving in this protocol -

  • One Coordinating site

  • One or more participating sites.

Two phase commit protocol

This protocol works in two phases - i) Voting phase and ii) Decision phase.

Phase 1: Obtaining decision or voting phase

Step 1: Coordinator site Ci asks all participants to prepare to commit transaction Ti.

  • Ci adds the records <prepareT> to the log and writes the log to stable storage.

  • It then sends prepare T messages to all participating sites at which T will get executed.

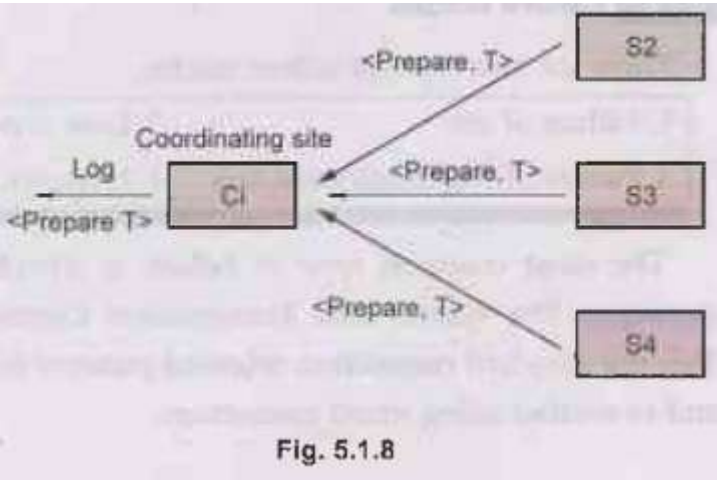


Fig. 5.1.8

Step 2: Upon receiving message, transaction manager at participating site determines if it can commit the transaction

• If not, add a record <no T> to the log and send abort T message to coordinating site Ci.

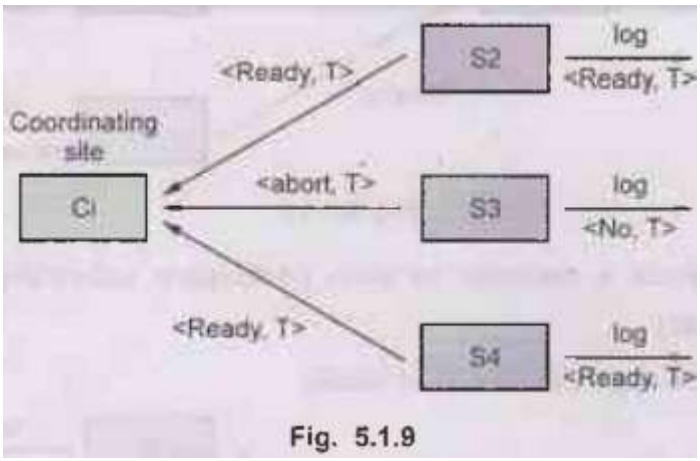


Fig. 5.1.9

• If the transaction can be committed, then :

  • Add the record <ready T> to the log

  • Force all records for T to stable storage

  • Send ready T message to Ci.

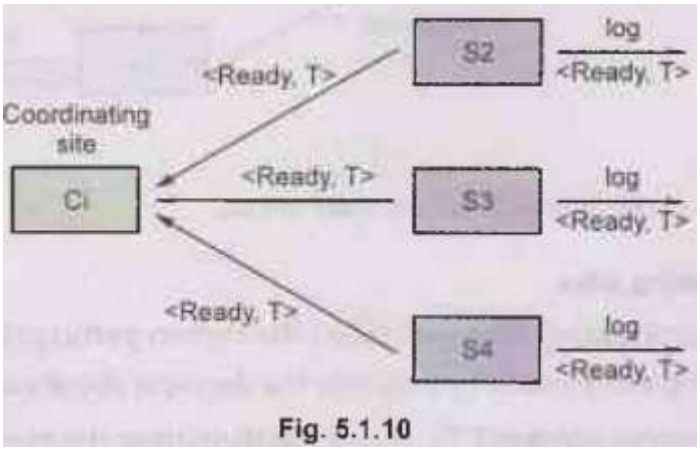


Fig. 5.1.10

Phase 2: Recoding decision phase

• T can be committed of Ci received a ready T message from all the participating sites otherwise T must be aborted.o

• Coordinator adds a decision record, <commit T> or <abort T>, to the log and forces record onto stable storage. Once the record stable storage it is irrevocable (even if failures occur)

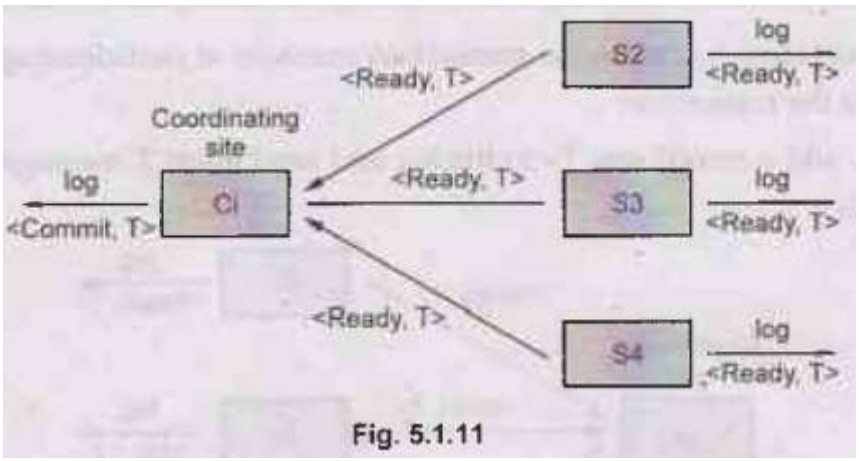
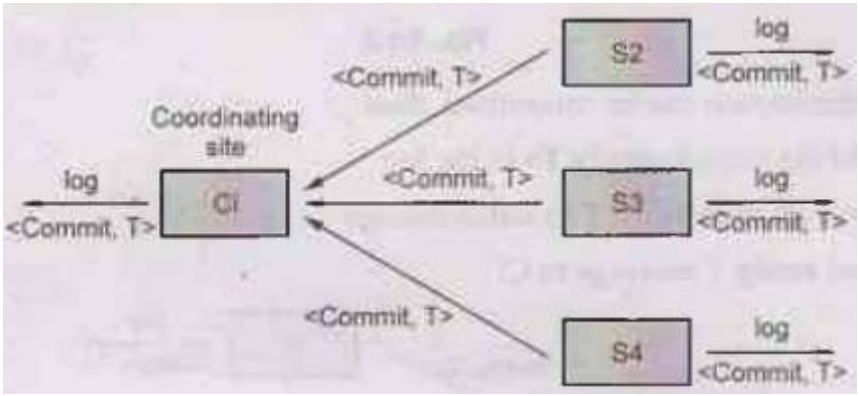


Fig. 5.1.11

• Coordinator sends a message to each participant informing it of the decision (commit or abort)

• Participants take appropriate action locally.



Failure of site

There are various cases at which failure may occur,

(1) Failure of participating sites

• If any of the participating sites gets failed then when participating site Si recovers, it examines the log entry made by it to take the decision about executing transaction.

• If the log contains <commit T> record: participating site executes redo (T)

• If the log contains <abort T> record: participating site executes undo (T)

• If the log contains <ready T> record: participating site must consult Coordinating site to take decision about execution of transaction T.

• If T committed, redo (T)

• If T aborted, undo (T)

• If the log of participating site contains no record then that means Si gets failed before responding to Prepare T message from coordinating site. In this case it must abort T

(2) Failure of coordinator

• If coordinator fails while the commit protocol for T is executing then participating sites must take decision about execution of transaction T:

i) If an active participating site contains a <commit T> record in its log, then T site must be committed.

ii) If an active participating site contains an <abort T> record in its log, then T must be aborted.

iii) If some active participating site does not contain a <ready T> record in its log, then the failed coordinator Ci cannot have decided to commit T. Can therefore abort T.

iv) If none of the above cases holds, then all participating active sites must have a <ready T> record in their logs, but no additional control records (such as <abort T> of <commit T>). In this case active sites must wait for coordinator site Ci to recover, to find decision.

Two phase locking protocol has blocking problem.

What is blocking problem?

It is a stage at which active participating sites may have to wait for failed coordinator site to recover.

The solution to this problem is to use three phase locking protocol.

Three Phase Commit Protocol

• The three phase locking is an extension of two phase locking protocol in which eliminates the blocking problem.

• Various assumptions that are made for three phase commit protocol are -

  • No network partitioning.

  • At any point at least one site must be up.

  • At the most k sites (participating as well as coordinating) can fail.

• **Phase 1:** This phase is similar to phase 1 of two phase protocol. That means Coordinator site Ci asks all participants to prepare to commit transaction Ti. The coordinator then makes the decision about commit or abort based on the response from all the participating sites.

• **Phase 2:** In phase 2 coordinator makes a decision as in 2 Phase Commit which is called the pre-commit decision <Pre-commit, T>, and records it in multiple (at least K) participating sites.

• **Phase 3:** In phase 3, coordinator sends commit/abort message to all participating Brits sites.

• Under three phase protocol, the knowledge of pre-commit decision can be used to commit despite coordinator site failure. That means if the coordinating site in case gets failed then one of the participating site becomes the coordinating site and der consults other participating sites to know the Pre-commit message which they possess. Thus using this pre-commit t message the decision about commit/abort is taken by this new coordinating site.

• This protocol avoids blocking problem as long as less than k sites fail.

Advantage of three phase commit protocol

(1) It avoid blocking problem.

Disadvantage of three phase commit protocol

(1) The overhead is increased.

Query Processing and Optimization

Distributed database query is processed using following steps-

(1) Query Mapping:

• The input query on distributed data is specified using query language.

• This query language is then translated into algebraic query.

• During this translation the global conceptual schema is referred.

• During the translation some important actions such as normalization, analysis for semantic errors, simplification are carried out then input query is restructured into algebraic query.

(2) Localization:

• In this step, the replication of information is handled.

• The distributed query is mapped on the global schema to separate queries on individual fragments.

(3) Global Query Optimization: optimization means selecting a strategy from list of candidate queries which is closest to optimal. For optimization, the cost is computed. The total cost is combination of CPU cost, I/O cost and communication costs.

(4) Local Query Optimization: This step is common to all sites in distributed database. The techniques of local query optimization are similar to those used in centralized systems.

NOSQL DATABASES

Introduction

• NoSQL stands for not only SQL.

• It is nontabular database system that store data differently than relational tables. There are various types of NoSQL databases such as document, key-value, wide column and graph.

• Using NoSQL we can maintain flexible schemas and these schemas can be scaled easily with large amount of data

Need

The NoSQL database technology is usually adopted for following reasons -ut

1) The NoSQL databases are often used for handling big data as a part of fundamental architecture.

2) The NoSQL databases are used for storing and modelling structured, semi-structured and unstructured data.

3) For the efficient execution of database with high availability, NoSQL is used.

4) The NoSQL database is non-relational, so it scales out better than relational databases and these can be designed with web applications.

5) For easy scalability, the NoSQL is used.

Features

1) The NoSQL does not follow any relational model.

2) It is either schema free or have relaxed schema. That means it does not require specific definition of schema.

3) Multiple NoSQL databases can be executed in distributed fashion.

4) It can process both unstructured and semi-structured data.

5) The NoSQL have higher scalability.

6) It is cost effective.

7) It supports the data in the form of key-value pair, wide columns and graphs.

Comparison between RDBMS and NoSQL

Sr. No.	RDBMS	NoSQL
1.	The relational database system is based on relationships among the tables.	It is non-relational database system. It can be used in distributed environment.
2.	It is vertically scalable.	It is horizontally scalable.
3.	It has predefined schema.	It does not have schema or it may have relaxed schema.
4.	It uses SQL to query the database.	It uses unstructured query language.
5.	It is a table based database.	It is document based, graph based or key-value pair.
6.	It emphasizes on ACID properties (Automicity, consistency, isolation and durability)	It follows Brewers CAP theorem (Consistency, availability and partition tolerance)
7.	Schema is fixed or rigid.	Schema is dynamic.
8.	Pessimistic.	Optimistic.
9.	Examples : MySQL,Oracle, PostgreSQL.	Examples : MangoDB, BigTable, Redis.

TYPES OF NOSQL DATABASE

There are four types of NoSQL databases and those are -

1. Key-value store

2. Document store

3. Graph based

4. Wide column store

Let us discuss them in detail.

Key-Value Store

• Key-value pair is the simplest type of NoSQL database.

• It is designed in such a way to handle lots of data and heavy load.

• In the key-value storage the key is unique and the value can be JSON, string or binary objects.

• For example -

{Customer:

|

{"id":1,"name":"Ankita"},

{"id":2,"name":"Kavita"}

|

}

Here id, name are the keys and 1,2, "Ankita", "Prajkta" are the values corresponding to those keys.

Key-value stores help the developer to store schema-less data. They work best for Shopping cart contents.

The DynamoDB, Riak, Redis are some famous examples of key-value store.

Document Based Systems

• The document store make use of key-value pair to store and retrieve data.

• The document is stored in the form of XML and JSON.

• The document stores appear the most natural among NoSQL database types.

• It is most commonly used due to flexibility and ability to query on any field.

• For example -

{

"id": 101,

"Name": "AAA",

"City" : "Pune"

}

MongoDB and CouchDB are two popular document oriented NoSQL database.

Column Based Systems

• The column store model is similar to traditional relational database. In this model, the columns are created for each row rather than having predefined by the table structure.

• In this model number of columns are not fixed for each record.

• Columns databases can quickly aggregate the value of a given column.

• For example -

Row ID	Columns...		
1	Name	City	
	Ankita	Pune	
2	Name	City	email
	Kavita	Mumbai	kavita123@gmail.com

The column store databases are widely used to manage data warehouses, business intelligence, HBase, Cassandra are examples of column based databases.

Graph Databases

The graph database is typically used in the applications where the relationships among the data elements is an important aspect.

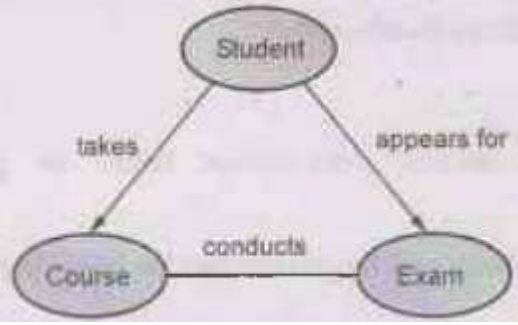
The connections between elements are called links or relationships. In a graph database, connections are first-class elements of the database, stored directly. In relational databases, links are implied, using data to express the relationships.

The graph database has two components -

1) Node: The entities itself. For example - People, student.

2) Edge: The relationships among the entities.

For example -



Graph base database is mostly used for social networks, logistics, spatial data. The graph databases are - Neo4J, Infinite Graph,OrientDB.

CAP THEOREM

Cap theorem is also called as brewer's theorem.

• The CAP theorem is comprised of three components (hence its name) as they relate to distributed data stores:

- **Consistency:** All reads receive the most recent write or an error.
- **Availability:** All reads contain data, but it might not be the most recent.

• **Partition tolerance:** The system continues to operate despite network failures(i.e.; dropped partitions, slow network connections, or unavailable network connections between nodes.)

• The CAP theorem states that it is not possible to guarantee all three of the desirable properties - Consistency, availability and partition tolerance at the same time in a distributed system with data replication.

DATABASE SECURITY

Definition: Database security is a technique that deals with protection of data against unauthorized access and protection.

• Database security is an important aspect for any database management system as it deals with sensitivity of data and information of enterprise.

• Database security allows or disallows users. from performing actions on the database objects.

Security Issues

Types of Security

Database security addresses following issues -

(1) Legal Issues: There are many legal or ethical issues with respect to right to access information. For example - If some sensitive information is present in the database, then it must not be accessed by unauthorized person.

(2) Policy Issues: There are some government or organizational policies that tells us what kind of information should be made available to access publicly.

(3) System Issues: Under this issue, it is decided whether security function should be handled at hardware level or at operating system level or at database level.

(4) Data and User Level Issues: In many organizations, multiple security levels are identified to categorize data and users based on these classifications. The security policy of organization must understand these levels for permitting access to different levels of users.

Threats to Database

Threats to database will result in loss or degradation of data. There are three kinds of loss that occur due to threats to database

(1) Loss of Integrity:

• Database integrity means information must be protected from improper modification.

• Modification to database can be performed by inserting, deleting or modifying the data.

• Integrity is lost if unauthorized changes are made to data intentionally or accidently.

• If data integrity is not corrected and work is continued then it results in inaccuracy, fraud, or erroneous decision.

(2) Loss of Availability:

• Database availability means making the database objects available to authorized

users.

(3) Loss of Confidentiality:

• Confidentiality means protection of data from unauthorized disclosure of information.

• The loss of confidentiality results in loss of public confidence, or embarrassment or some legal action against organization.

Control Measures

• There are four major control measures used to provide security on data in database.

1. Access control

2. Interface control

3. Flow control

4. Data encryption

• **Access Control:** The most common security problem is unauthorized access to of computer system. Generally this access is for obtaining the information or to make malicious changes in the database. The security mechanism of a DBMS must include provisions for restricting access to the database system as a whole. This function, called access control.

• **Inference Control:** This method is used to provide the security to statistical database security problems. Statistical databases are used to provide statistical information based on some criteria. These databases may contain information about particular age group, income-level, education criteria and so on. Access to some sensitive information must be avoided while using the statistical databases. The corresponding measure that prevents the user from completing any inference channel.

• **Flow Control:** It is a kind of control measure which prevents information from flowing in such a way that it reaches unauthorized users. Channels that are pathways for information to flow implicitly in ways that violate the security policy of an organization are called covert channels.

• **Data Encryption:** The data encryption is a control measure used to secure the sensitive data. In this technique, the data is encoded using some coding algorithm. An unauthorized user who accesses encoded data will have difficulty deciphering it, but authorized users are given decoding or decrypting algorithms (or keys) to decipher the data.

Database Security and DBA

• DBA stands for Database Administrator, who is the central authority for managing the database system.

• DBA is responsible for granting privileges to users who want to use the database system.

• The DBA has a DBA account in the DBMS which is sometimes called as system or superuser account. It provides powerful capabilities that are not made available for regular database accounts and users.

• DBA makes use of some special commands that perform following type of actions –

1. Account Creation: This command helps in creating a new account and password for a single user or for group of users.

2. Privilege Granting: This command allows the DBA to grant privileges to certain accounts.

3. Privilege Revocation: This command allows the DBA to cancel the privileges to certain accounts.

4. Security Level Assignment: This action assigns user account to the appropriate security clearance level.

Thus the DBA is responsible for the overall security of the database system.

ACCESS CONTROL BASED ON PRIVILEGES

Access Control Based on Privileges or Discretionary Access Control

• Discretionary Access Control (DAC) is a access control mechanism based on privileges.

• **Types of discretionary privileges**: The DBMS must provide selective access to each relation in the database on specific accounts. This selective access is known as privileges. There are two levels for assigning privileges for using Database systems and these are -

• The account level: At this level, the DBA specifies the particular privileges that each account holds independently of the relations in the database.

• Relation(or table) level: At this level, the DBA can control the privilege to access each individual relation or view in the database.

• For granting the privileges, the access control mechanism follows an authorizationof (a model for discretionary privileges known as the access matrix model.

• The access matrix is a table with rows and columns. It defines the access permissions.

• The rows of a matrix M represent subjects (users, accounts, programs)

• The columns represent objects (relations, records, columns, views, operations).

• Each position M(i, j) in the matrix represents the types of privileges (read, write, esdstab update) that subject i holds on object j.

• For example -

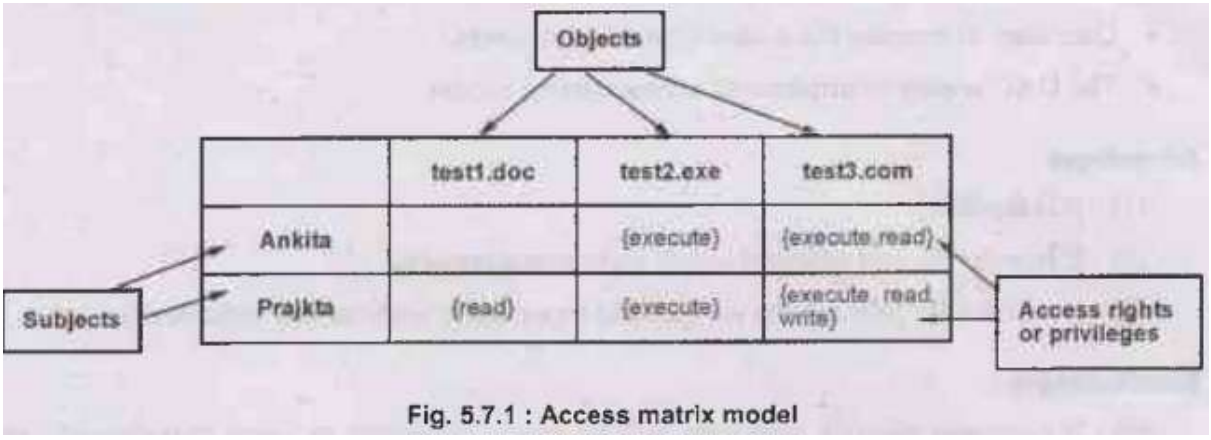


Fig. 5.7.1 : Access matrix model

• Discretionary Access Control allows each user or subject to control access to their own data.

• In DAC, owner of resource restricts access to the resources based on the identity of users.

• DAC is typically the default access control mechanism for most desktop operating doy systems.

• Each resource object on DAC based system has Account Control List (ACL) associated with it.

• An ACL contains a list of users and groups to which the user has permitted access together with the level of access for each user or group.

• For example - The ACL is an object centered description of access rights as follows-

test1.doc: {Prajka: read}

test2.exe: {Ankita: execute}, {Prajkta: execute}

test3.com: (Ankita: execute, read}, {Prajkta: execute, read, write}

• Object access is determined during Access Control List (ACL) authorization and based on user identification and/or group membership.

• Under DAC a user can only set access permissions for resources which they already own.

• Similarly a hypothetical user A cannot change the access control for a file that is owned by user B. User A can, however, set access permissions on a file that he/she Jono owns.

• User may transfer object ownership to another user(s).

• User may determine the access type of other users.

• The DAC is easy to implement access control model.

Advantages:

(1) It is flexible.

(2) It has simple and efficient access right management.

(3) It is scalable. That means we can add more users without any complexity.

Disadvantages:

(1) It increases the risk that data will be made accessible to users that should not necessarily be given access.

(2) There is no control over information flow as one user can transfer ownership to another user.

ROLE BASED ACCESS CONTROL

It is based on the concept that privileges and other permissions are associated with A organizational roles, rather than individual users. Individual users are then assigned to appropriate roles.

• For example, an accountant in a company will be assigned to the Accountant role, gaining access to all the resources permitted for all accountants on the system. Similarly, a software engineer might be assigned to the Developer role.

• In an RBAC system, the roles are centrally managed by the administrator. The administrators determine what roles exist within their companies and then map these roles to job functions and tasks.

• Roles can effectively be implemented using security groups. The security groups are created representing each role. Then permissions and rights are assigned to these groups. Next, simply add the appropriate users to the appropriate security groups, depending on their roles or job functions.

• A user can have more than one role. And more than one user can have the same role.

• Role hierarchies can be used to match natural relations between roles. For example - A lecturer can create a role student and give it a privilege "read course material".

• Role Based Access Control (RBAC), also known as non discretionary access control.

• RBAC security strategy is widely used by most organizations for deployment of commercial and off-the-shelf products.

Advantages:

(1) The security is more easily maintained by limiting unnecessary access to sensitive information based on each user's established role within the organization.

(2) All the roles can be aligned with the organizational structure of the business and users can do their jobs more efficiently and autonomously.

Disadvantages:

(1) It is necessary to understand each user's functionality in depth so that roles can be properly assigned.

(2) If roles are not assigned properly then inappropriate access right creates security severe problems for database system.

SQL INJECTION

• SQL injection is a type of code injection technique that might destroy the databases.

• In this technique the malicious code in SQL statement is placed via web page input. These statements control a database server behind a web application.

• Attackers can use SQL injection vulnerabilities to bypass application security measures. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQL injection to add, modify and delete records in the database.

• An SQL injection vulnerability may affect any website or web application that uses an SQL database such as MySQL, Oracle, SQL Server or others.

How SQL Injection Works?

• To make an SQL injection attack, an attacker must first find vulnerable user inputs ad to within the web page or web application. A web page or web application that has an ses SQL injection vulnerability uses such user input directly in an SQL query. The attacker can create input content. Such content is often called a malicious

payload and is the key part of the attack. After the attacker sends this content, malicious SQL commands are executed in the database.

• SQL is a query language that was designed to manage data stored in relational Sup databases. You can use it to access, modify and delete data. Many web applications and websites store all the data in SQL databases. In some cases, you can also use SQL commands to run operating system commands. Therefore, a successful SQL Injection attack can have very serious consequences.

Example of SQL Injection

• Following is an example of SQL injection vulnerability works around a simple rid web application having two input fields - One for user name and another for password.

• This example has a table named users with the columns username and password

uname-request.POST['username']

passwd=request.POST['password']

query="SELECT id FROM users WHERE username='"+ uname +"' ANDpassword='"+ passwd +"'

database.execute(query)

• Here the two input fields - One for user name and another for password is vulnerable to SQL injection.

• The attacker can attack using these fields and alter the SQL query to get the access to the database.

• They could use a trick on password field. They could add

OR 1 = 1

Statement to the password field.

• As a result the query would becomes (assuming username as 'user1' and password='password')

• SELECT id FROM users WHERE username='user1' AND password='password' OR 1 = 1

• Because of OR 1 = 1 statement, the WHERE clause returns the first id from the users table no matter what the username and password are. That means even-if we enter any wrong username or password still the query will get executed because of OR 1 = 1 part which comes out to be true.

• The first id is returned by the above query for users table and we know that the first id is normally administrator. In this way, the attacker not only bypasses authentication but also gains administrator privileges.

How to prevent SQL injection?

• The only way to prevent SQL injection is to validate every input field.

• Another method is to make use of parameterized query. This parameterized query is called prepared statement. By this ways, application code never use the input directly.

• The Web Application Firewalls (WAF) are also used to filter out the SQL.

ENCRYPTION AND PUBLIC KEY INFRASTRUCTURES

Cryptology is a technique of encoding and decoding messages, so that they cannot be understood by anybody except the sender and the intended recipient.

There are various encoding and decoding schemes which are called as encryption schemes. The sender and recipient of the message decide on an encoding and decoding scheme and use it for communication.

The process of encoding messages is known as encryption. The sender sends the original text. The original text called plaintext, The encrypted form of plaintext it is called as ciphertext. This encrypted text travel through the network. When it reaches at the receiving computer, the recipient understands the meaning and decodes the message to extract the correct meaning out of it. This process is called as decryption.

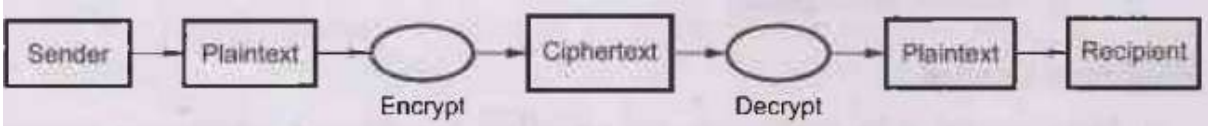


Fig. 5.12.1 : Encryption and decryption process

The sender applies the encryption algorithm and recipient applies the decryption algorithm. Both the sender and the receiver must agree on this algorithm for any meaningful communication. The algorithm basically takes one text as input and produces another as the output. Therefore, the algorithm contains the intelligence for transforming message.

For example: If we want to send some message through an e-mail and we wish that nobody except the friend should be able to understand it. Then the message can be encoded using some intelligence. For example if the alphabets A to Z are encoded as follows-

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Z	Y	X	A	B	C	W	V	U	D	E	F	T	S	R	G	H	I	Q	P	O	J	K	L	N	M

That means last three letters are placed in reverse order and then first three letters are in straight manner. Continuing this logic the A to Z letters are encoded. Now if I write the message

"SEND SOME MONEY"

it will be

QBSA QRTB TRSBN

This coded message is called cipher text.

There are variety of coding methods that can be used.

Types of Cryptography

There are two types encryption schemes based in key used for encryption and decryption.

1. Symmetric key encryption: It is also known as secret key encryption. In this method, only one key is used. The same key is shared by sender and receiver for encryption and decryption of messages. Hence both parties must agree upon the key before any transmission begins and nobody else should know about it. At the sender's end, the key is used to change the original message into an encoded form. At the receiver's end using the same key the encoded message is decrypted and original message is obtained. Data Encryption Standard (DES) uses this approach. The problem with this approach is that of key agreement and distribution.

2. Asymmetric key encryption: It is also known as public key encryption. In this method, different keys are used. One key is used for encryption and other key must be used for decryption. No other key can decrypt the message-not even the original key used for encryption.

One of the two keys is known as public key and the other is the private key. Suppose there are two users X and Y. The

• X wants to send a message to Y. Then X will convey its public key to Y but the private key of X will be known to X only.

• Y should know the private key of Y and X should know the Y's public key.

When X and Y wants to communicate:m

1. If X wants to send a message to Y, then first of all X encrypts the message using Y's public key. For that purpose it is necessary that X knows the Y's public key.

2. X then sends this encrypted to Y.

3. Now using Y's private key, Y decrypts X's message. Note that only Y knows his private key. It is not possible for Y to decrypt the message using X's public key.

4. When Y wants to send a message to X then using X's public key Y will encrypt the message and will send the encrypted message to X. On the other hand, X will use its own private key to decrypt this message. Here again Y will not know the private key of X.

Digital Signature

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. If the recipient gets a message with digital signature then he believes that the message was created by a known sender.

Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering.

When X and Y wants to communicate with each other

1. X encrypts the original plaintext message into ciphertext by using Y's public key.

2. Then X executes an algorithm on the original plaintext to calculate a Message Digest, also known as hash. This algorithm takes the original plaintext in the binary format, apply the hashing algorithm. As an output a small string of binary digits gets created. This hashing algorithm is public and anyone can use it. The most popular message digest algorithms are MD5 and SHA-1. X encrypts the message digest. For this, it uses its own private key.

3. X now combines the ciphertext and its digital signature (i.e encrypted message digest) and it is sent over the network to Y.

4. Y receives the ciphertext and X's digital signature. Y has to decrypt both of these. Y first decrypts ciphertext back to plaintext. For this, it uses its own private key. Thus, Y gets the message itself in a secure manner.

5. Now to ensure that the message has come from the intended sender Y takes X's digital signature and decrypts it. This gives Y the message digest as was generated by X. The X had encrypted the message digest to form a digital signature using its own private key. Therefore, Y uses X's public key for decrypting the digital signature.

6. Hash algorithm to generate the message digest is public. Therefore, Y can also use it.

7. Now there are two message digests one created by X and other by Y. The Y now Anon simply compares the two message digests. If the two match, Y can be sure that the message came indeed from X and not from someone else.

Thus with digital signature confidentiality, authenticity as well as message integrity is assured.

The other important feature supported by digital signature is non-repudiation. That is, a sender cannot refuse having sent a message. Since the digital signature requires the private key of the sender, once a message is digitally signed, it can be legally proven that the sender had indeed sent the message.

CHALLENGES

Following are the challenges faced by the database security system -

(1) Data Quality

• The database community need the solution to assess the quality of data. The quality of data can be assessed by a simple mechanism such as quality stamps that are posted on web sites:

• The database community may need more effective technique of integrity semantic verification for accessing the quality of data.

• Application level recovery techniques are also used to repair incorrect data.

(2) Intellectual Property Rights

• Everywhere there is increasing use of internet and intranet. Due to which there are chances of making un-authorized duplication and distribution of the contents. Hence digital watermarking technique is used to protect the contents from unauthorized access or ownership.

• However, research is needed to develop the techniques for preventing intellectual property right violation.

(3) Database Survivability

• It is desired that the database systems must continue to work even after information warfare attacks.

• The goal of information warfare attacker is to damage the organization's operation.

• Following are the corrective actions for handling this situation -

• **Confinement :** Take immediate action to eliminate attacker's access to the system. Isolate the affected components to avoid further spread.

• **Damage Assessment:** Determine the extent of problem.

• **Reconfiguration:**Re-configuration allows the system to be in operation in degraded mode while recovery is going on.

• **Repair:**Recover the corrupted or lost data by repairing or reinstalling the system.

• **Fault treatment:** Identify the weakness exploited in the attack and take steps to prevent a recurrence.