

MODUL PRAKTIKUM PERTEMUAN 9 - 15

KODE MATA KULIAH CPD121

MATA KULIAH PEMROSESAN DATA TERSEBAR (PDT)

OLEH

Ir. Munawar, MMSI, M.Com, Ph.D

Ir. Nizirwan Anwar, MT

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS ESA UNGGUL**

2018

DAFTAR ISI
MODUL PRAKTIKUM

Modul 9 *Consistency-and-Replication*

Modul 10 *Fault-Tolerance*

Modul 11 *Security*

Modul 12 *Distributed-Object-System*

Modul 13 *Distributed File System*

Modul 14 *Distributed Web Based System*

Modul 15 *Distribution Coordinator Based System*

DAFTAR PUSTAKA

Ajay D. Kshemkalyani and Mukesh Singhal, (2008), Distributed Computing - Principles, Algorithms, and Systems, Cambridge University Press, ISBN-13 978-0-511-39341-9

CognosTutorial – Simple Easy Learning, (2016),
https://www.tutorialspoint.com/cognos/cognos_tutorial.pdf di-akses tanggal 5 Maret 2018 di Jakarta

Coulouris et. al. (2012), Distributed Systems Concepts and Design, Fifth Edition. Addison Wesley, ISBN 13: 978-0-13-214301-1

Maarten van Steen and Andrew S. Tanenbaum, (2017), Distributed Systems : Prinsiples and Paradigms. 3th Pearson Education Inc.

P M. Tamer Özsu and Patrick Valduriez, (2011), Principles of Distributed Database Systems, Third Edition, Springer Publishing ISBN 978-1-4419-8833-1



Modul 9

Consistency-and-Replication



19. MongoDB – Replication

Replication is the process of synchronizing data across multiple servers. Replication provides redundancy and increases data availability with multiple copies of data on different database servers. Replication protects a database from the loss of a single server. Replication also allows you to recover from hardware failure and service interruptions. With additional copies of the data, you can dedicate one to disaster recovery, reporting, or backup.

Why Replication?

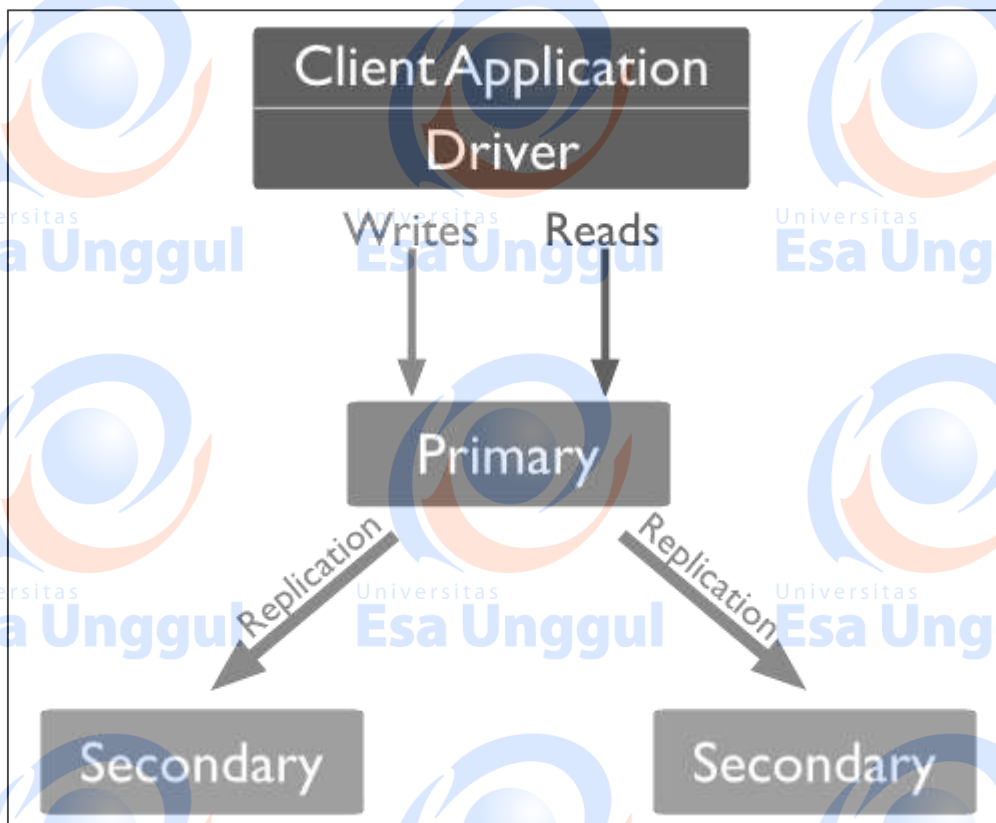
- To keep your data safe
- High (24*7) availability of data
- Disaster recovery
- No downtime for maintenance (like backups, index rebuilds, compaction)
- Read scaling (extra copies to read from)
- Replica set is transparent to the application

How Replication Works in MongoDB

MongoDB achieves replication by the use of replica set. A replica set is a group of **mongod** instances that host the same data set. In a replica, one node is primary node that receives all write operations. All other instances, such as secondaries, apply operations from the primary so that they have the same data set. Replica set can have only one primary node.

- Replica set is a group of two or more nodes (generally minimum 3 nodes are required).
- In a replica set, one node is primary node and remaining nodes are secondary.
- All data replicates from primary to secondary node.
- At the time of automatic failover or maintenance, election establishes for primary and a new primary node is elected.
- After the recovery of failed node, it again joins the replica set and works as a secondary node.

A typical diagram of MongoDB replication is shown in which client application always interact with the primary node and the primary node then replicates the data to the secondary nodes.



Replica Set Features

- A cluster of N nodes
- Any one node can be primary
- All write operations go to primary
- Automatic failover
- Automatic recovery
- Consensus election of primary

Set Up a Replica Set

In this tutorial, we will convert standalone MongoDB instance to a replica set. To convert to replica set, following are the steps:

- Shutdown already running MongoDB server.
- Start the MongoDB server by specifying `-- replSet` option. Following is the basic syntax of `--replSet`:

```
mongod --port "PORT" --dbpath "YOUR_DB_DATA_PATH" --replSet
"REPLICA_SET_INSTANCE_NAME"
```

Example

```
mongod --port 27017 --dbpath "D:\set up\mongodb\data" --replSet rs0
```

- It will start a mongod instance with the name rs0, on port 27017.
- Now start the command prompt and connect to this mongod instance.
- In Mongo client, issue the command **rs.initiate()** to initiate a new replica set.
- To check the replica set configuration, issue the command **rs.conf()**. To check the status of replica set issue the command **rs.status()**.

Add Members to Replica Set

To add members to replica set, start mongod instances on multiple machines. Now start a mongo client and issue a command **rs.add()**.

Syntax

The basic syntax of **rs.add()** command is as follows:

```
>rs.add(HOST_NAME:PORT)
```

Example

Suppose your mongod instance name is **mongod1.net** and it is running on port **27017**. To add this instance to replica set, issue the command **rs.add()** in Mongo client.

```
>rs.add("mongod1.net:27017")
>
```

You can add mongod instance to replica set only when you are connected to primary node. To check whether you are connected to primary or not, issue the command **db.isMaster()** in Mongo client.

20. MongoDB – Sharding

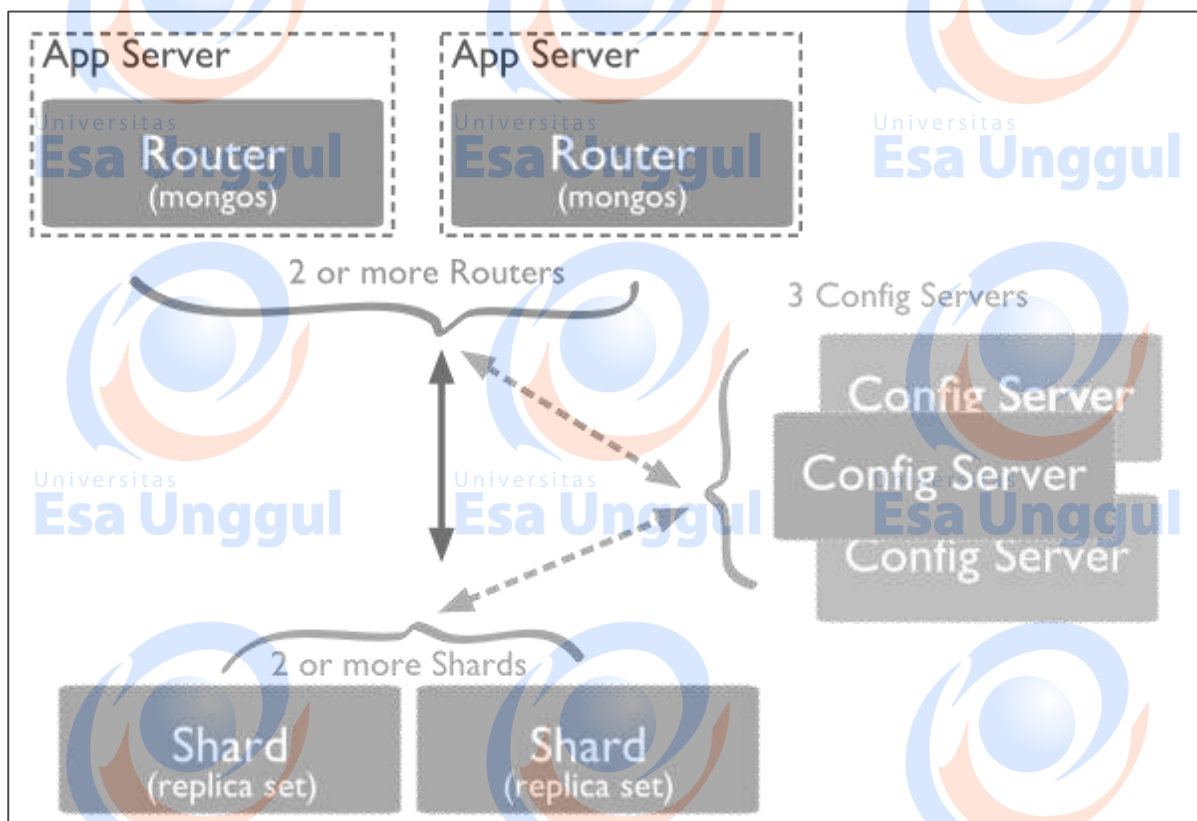
Sharding is the process of storing data records across multiple machines and it is MongoDB's approach to meeting the demands of data growth. As the size of the data increases, a single machine may not be sufficient to store the data nor provide an acceptable read and write throughput. Sharding solves the problem with horizontal scaling. With sharding, you add more machines to support data growth and the demands of read and write operations.

Why Sharding?

- In replication, all writes go to master node
- Latency sensitive queries still go to master
- Single replica set has limitation of 12 nodes
- Memory can't be large enough when active dataset is big
- Local disk is not big enough
- Vertical scaling is too expensive

Sharding in MongoDB

The following diagram shows the sharding in MongoDB using sharded cluster.



In the following diagram, there are three main components:

- **Shards:** Shards are used to store data. They provide high availability and data consistency. In production environment, each shard is a separate replica set.
- **Config Servers:** Config servers store the cluster's metadata. This data contains a mapping of the cluster's data set to the shards. The query router uses this metadata to target operations to specific shards. In production environment, sharded clusters have exactly 3 config servers.
- **Query Routers:** Query routers are basically mongo instances, interface with client applications and direct operations to the appropriate shard. The query router processes and targets the operations to shards and then returns results to the clients. A sharded cluster can contain more than one query router to divide the client request load. A client sends requests to one query router. Generally, a sharded cluster have many query routers.

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

DISTRIBUTED DBMS - REPLICATION CONTROL

https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_replication_control.htm

Copyright © tutorialspoint.com

This chapter looks into replication control, which is required to maintain consistent data in all sites. We will study the replication control techniques and the algorithms required for replication control.

As discussed earlier, **replication** is a technique used in distributed databases to store multiple copies of a data table at different sites. The problem with having multiple copies in multiple sites is the overhead of maintaining data consistency, particularly during update operations.

In order to maintain mutually consistent data in all sites, replication control techniques need to be adopted. There are two approaches for replication control, namely –

- Synchronous Replication Control
- Asynchronous Replication Control

Synchronous Replication Control

In synchronous replication approach, the database is synchronized so that all the replications always have the same value. A transaction requesting a data item will have access to the same value in all the sites. To ensure this uniformity, a transaction that updates a data item is expanded so that it makes the update in all the copies of the data item. Generally, two-phase commit protocol is used for the purpose.

For example, let us consider a data table PROJECTPId, PName, PLocation. We need to run a transaction T1 that updates PLocation to 'Mumbai', if PLocation is 'Bombay'. If no replications are there, the operations in transaction T1 will be –

```
Begin T1:
  Update PROJECT Set PLocation = 'Mumbai'
  Where PLocation = 'Bombay';
End T1;
```

If the data table has two replicas in Site A and Site B, T1 needs to spawn two children T1A and T1B corresponding to the two sites. The expanded transaction T1 will be –

```
Begin T1:
  Begin T1A :
    Update PROJECT Set PLocation = 'Mumbai'
    Where PLocation = 'Bombay';
  End T1A;

  Begin T2A :
    Update PROJECT Set PLocation = 'Mumbai'
    Where PLocation = 'Bombay';
  End T2A;

End T1;
```

Asynchronous Replication Control

In asynchronous replication approach, the replicas do not always maintain the same value. One or more replicas may store an outdated value, and a transaction can see the different values. The process of bringing all the replicas to the current value is called **synchronization**.

A popular method of synchronization is store and forward method. In this method, one site is designated as the primary site and the other sites are secondary sites. The primary site always contains updated values. All the transactions first enter the primary site. These transactions are then queued for application in the secondary sites. The secondary sites are updated using rollout method only when a transaction is scheduled to execute on it.

Replication Control Algorithms

Some of the replication control algorithms are –

- Master-slave replication control algorithm.
- Distributed voting algorithm.
- Majority consensus algorithm.
- Circulating token algorithm.

Master-Slave Replication Control Algorithm

There is one master site and 'N' slave sites. A master algorithm runs at the master site to detect conflicts. A copy of slave algorithm runs at each slave site. The overall algorithm executes in the following two phases –

- **Transaction acceptance/rejection phase** – When a transaction enters the transaction monitor of a slave site, the slave site sends a request to the master site. The master site checks for conflicts. If there aren't any conflicts, the master sends an "ACK+" message to the slave

site which then starts the transaction application phase. Otherwise, the master sends an "ACK-" message to the slave which then rejects the transaction.

- **Transaction application phase** – Upon entering this phase, the slave site where transaction has entered broadcasts a request to all slaves for executing the transaction. On receiving the requests, the peer slaves execute the transaction and send an "ACK" to the requesting slave on completion. After the requesting slave has received "ACK" messages from all its peers, it sends a "DONE" message to the master site. The master understands that the transaction has been completed and removes it from the pending queue.

Distributed Voting Algorithm

This comprises of 'N' peer sites, all of whom must "OK" a transaction before it starts executing. Following are the two phases of this algorithm –

- **Distributed transaction acceptance phase** – When a transaction enters the transaction manager of a site, it sends a transaction request to all other sites. On receiving a request, a peer site resolves conflicts using priority based voting rules. If all the peer sites are "OK" with the transaction, the requesting site starts application phase. If any of the peer sites does not "OK" a transaction, the requesting site rejects the transaction.
- **Distributed transaction application phase** – Upon entering this phase, the site where the transaction has entered, broadcasts a request to all slaves for executing the transaction. On receiving the requests, the peer slaves execute the transaction and send an "ACK" message to the requesting slave on completion. After the requesting slave has received "ACK" messages from all its peers, it lets the transaction manager know that the transaction has been completed.

Majority Consensus Algorithm

This is a variation from the distributed voting algorithm, where a transaction is allowed to execute when a majority of the peers "OK" a transaction. This is divided into three phases –

- **Voting phase** – When a transaction enters the transaction manager of a site, it sends a transaction request to all other sites. On receiving a request, a peer site tests for conflicts using voting rules and keeps the conflicting transactions, if any, in pending queue. Then, it sends either an "OK" or a "NOT OK" message.
- **Transaction acceptance/rejection phase** – If the requesting site receives a majority "OK" on the transaction, it accepts the transaction and broadcasts "ACCEPT" to all the sites. Otherwise, it broadcasts "REJECT" to all the sites and rejects the transaction.
- **Transaction application phase** – When a peer site receives a "REJECT" message, it removes this transaction from its pending list and reconsiders all deferred transactions. When a peer site receives an "ACCEPT" message, it applies the transaction and rejects all the deferred transactions in the pending queue which are in conflict with this transaction. It sends an "ACK" to the requesting slave on completion.

Circulating Token Algorithm

In this approach the transactions in the system are serialized using a circulating token and executed accordingly against every replica of the database. Thus, all the transactions are accepted, i.e. none is rejected. This has two phases –

- **Transaction serialization phase** – In this phase, all transactions are scheduled to run in a serialization order. Each transaction in each site is assigned a unique ticket from a sequential series, indicating the order of transaction. Once a transaction has been assigned a ticket, it is broadcasted to all the sites.
- **Transaction application phase** – When a site receives a transaction along with its ticket, it places the transaction for execution according to its ticket. After the transaction has finished execution, this site broadcasts an appropriate message. A transaction ends when it has completed execution in all the sites.



Modul 10

Fault-Tolerance



DISTRIBUTED DBMS - FAILURE & COMMIT

https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_failure_commit.htm

Copyright © tutorialspoint.com

A database management system is susceptible to a number of failures. In this chapter we will study the failure types and commit protocols. In a distributed database system, failures can be broadly categorized into soft failures, hard failures and network failures.

Soft Failure

Soft failure is the type of failure that causes the loss in volatile memory of the computer and not in the persistent storage. Here, the information stored in the non-persistent storage like main memory, buffers, caches or registers, is lost. They are also known as system crash. The various types of soft failures are as follows –

- Operating system failure.
- Main memory crash.
- Transaction failure or abortion.
- System generated error like integer overflow or divide-by-zero error.
- Failure of supporting software.
- Power failure.

Hard Failure

A hard failure is the type of failure that causes loss of data in the persistent or non-volatile storage like disk. Disk failure may cause corruption of data in some disk blocks or failure of the total disk. The causes of a hard failure are –

- Power failure.
- Faults in media.
- Read-write malfunction.
- Corruption of information on the disk.
- Read/write head crash of disk.

Recovery from disk failures can be short, if there is a new, formatted, and ready-to-use disk on reserve. Otherwise, duration includes the time it takes to get a purchase order, buy the disk, and prepare it.

Network Failure

Network failures are prevalent in distributed or network databases. These comprises of the errors induced in the database system due to the distributed nature of the data and transferring data over the network. The causes of network failure are as follows –

- Communication link failure.
- Network congestion.
- Information corruption during transfer.
- Site failures.
- Network partitioning.

Commit Protocols

Any database system should guarantee that the desirable properties of a transaction are maintained even after failures. If a failure occurs during the execution of a transaction, it may happen that all the changes brought about by the transaction are not committed. This makes the database inconsistent. Commit protocols prevent this scenario using either transaction undo *rollback* or transaction redo *rollforward*.

Commit Point

The point of time at which the decision is made whether to commit or abort a transaction, is known as commit point. Following are the properties of a commit point.

- It is a point of time when the database is consistent.
- At this point, the modifications brought about by the database can be seen by the other transactions. All transactions can have a consistent view of the database.
- At this point, all the operations of transaction have been successfully executed and their effects have been recorded in transaction log.
- At this point, a transaction can be safely undone, if required.
- At this point, a transaction releases all the locks held by it.

Transaction Undo

The process of undoing all the changes made to a database by a transaction is called transaction undo or transaction rollback. This is mostly applied in case of soft failure.

Transaction Redo

The process of reapplying the changes made to a database by a transaction is called transaction redo or transaction roll forward. This is mostly applied for recovery from a hard failure.

Transaction Log

A transaction log is a sequential file that keeps track of transaction operations on database items. As the log is sequential in nature, it is processed sequentially either from the beginning or from the end.

Purposes of a transaction log –

- To support commit protocols to commit or support transactions.
- To aid database recovery after failure.

A transaction log is usually kept on the disk, so that it is not affected by soft failures. Additionally, the log is periodically backed up to an archival storage like magnetic tape to protect it from disk failures as well.

Lists in Transaction Logs

The transaction log maintains five types of lists depending upon the status of the transaction. This list aids the recovery manager to ascertain the status of a transaction. The status and the corresponding lists are as follows –

- A transaction that has a transaction start record and a transaction commit record, is a committed transaction – maintained in commit list.
- A transaction that has a transaction start record and a transaction failed record but not a transaction abort record, is a failed transaction – maintained in failed list.
- A transaction that has a transaction start record and a transaction abort record is an aborted transaction – maintained in abort list.
- A transaction that has a transaction start record and a transaction before-commit record is a before-commit transaction, i.e. a transaction where all the operations have been executed but not committed – maintained in before-commit list.
- A transaction that has a transaction start record but no records of before-commit, commit, abort or failed, is an active transaction – maintained in active list.

Immediate Update and Deferred Update

Immediate Update and Deferred Update are two methods for maintaining transaction logs.

In **immediate update** mode, when a transaction executes, the updates made by the transaction are written directly onto the disk. The old values and the updates values are written onto the log before writing to the database in disk. On commit, the changes made to the disk are made permanent. On rollback, changes made by the transaction in the database are discarded and the old values are restored into the database from the old values stored in the log.

In **deferred update** mode, when a transaction executes, the updates made to the database by the transaction are recorded in the log file. On commit, the changes in the log are written onto the disk. On rollback, the changes in the log are discarded and no changes are applied to the database.



Modul 11

Security



16. DWH – Security

The objective of a data warehouse is to make large amounts of data easily accessible to the users, hence allowing the users to extract information about the business as a whole. But we know that there could be some security restrictions applied on the data that can be an obstacle for accessing the information. If the analyst has a restricted view of data, then it is impossible to capture a complete picture of the trends within the business.

The data from each analyst can be summarized and passed on to management where the different summaries can be aggregated. As the aggregations of summaries cannot be the same as that of the aggregation as a whole, it is possible to miss some information trends in the data unless someone is analyzing the data as a whole.

Security Requirements

Adding security features affect the performance of the data warehouse, therefore it is important to determine the security requirements as early as possible. It is difficult to add security features after the data warehouse has gone live.

During the design phase of the data warehouse, we should keep in mind what data sources may be added later and what would be the impact of adding those data sources. We should consider the following possibilities during the design phase.

- Whether the new data sources will require new security and/or audit restrictions to be implemented?
- Whether the new users added who have restricted access to data that is already generally available?

This situation arises when the future users and the data sources are not well known. In such a situation, we need to use the knowledge of business and the objective of data warehouse to know likely requirements.

The following activities get affected by security measures:

- User access
- Data load
- Data movement
- Query generation

User Access

We need to first classify the data and then classify the users on the basis of the data they can access. In other words, the users are classified according to the data they can access.

Data Classification

The following two approaches can be used to classify the data:

- Data can be classified according to its sensitivity. Highly-sensitive data is classified as highly restricted and less-sensitive data is classified as less restrictive.

- Data can also be classified according to the job function. This restriction allows only specific users to view particular data. Here we restrict the users to view only that part of the data in which they are interested and are responsible for.

There are some issues in the second approach. To understand, let's have an example. Suppose you are building the data warehouse for a bank. Consider that the data being stored in the data warehouse is the transaction data for all the accounts. The question here is, who is allowed to see the transaction data. The solution lies in classifying the data according to the function.

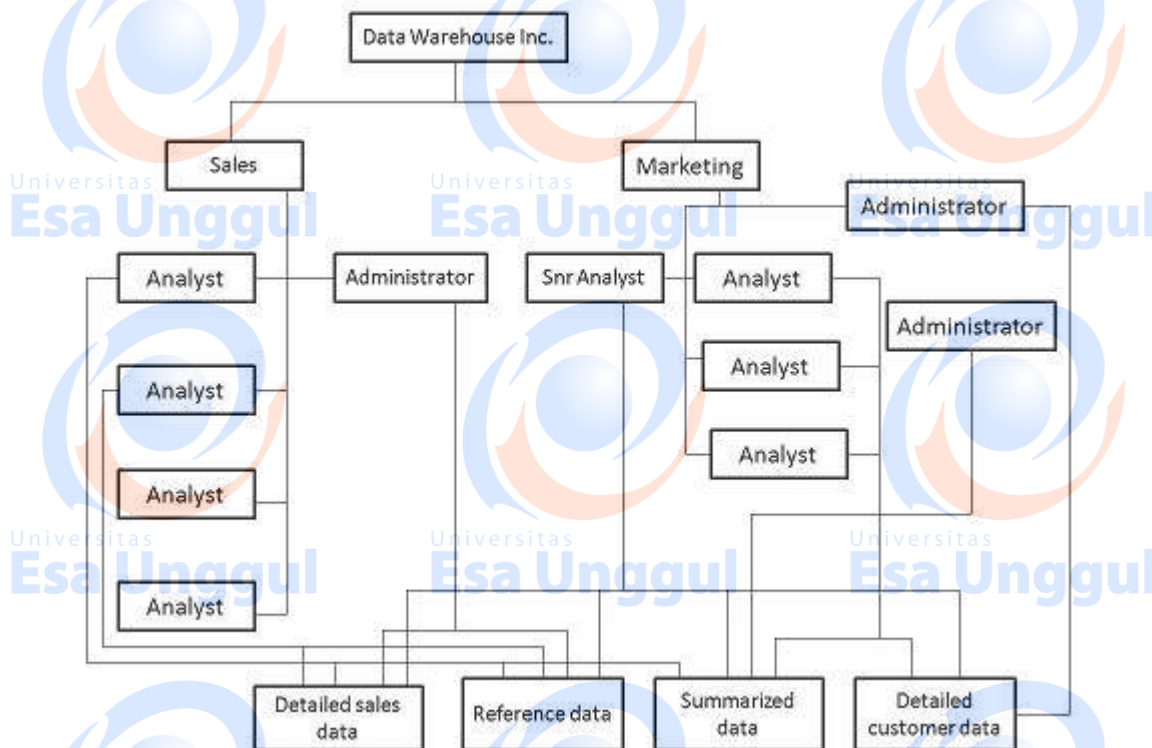
User Classification

The following approaches can be used to classify the users:

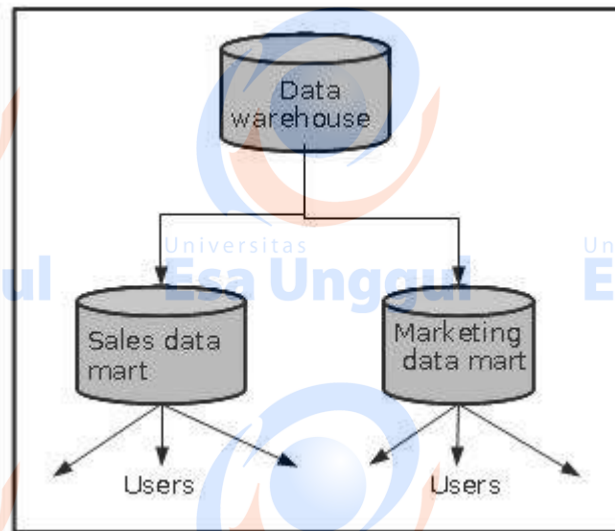
- Users can be classified as per the hierarchy of users in an organization, i.e., users can be classified by departments, sections, groups, and so on.
- Users can also be classified according to their role, with people grouped across departments based on their role.

Classification Based on Department

Let's have an example of a data warehouse where the users are from sales and marketing department. We can have security by top-to-down company view, with access centered on the different departments. But there could be some restrictions on users at different levels. This structure is shown in the following diagram.

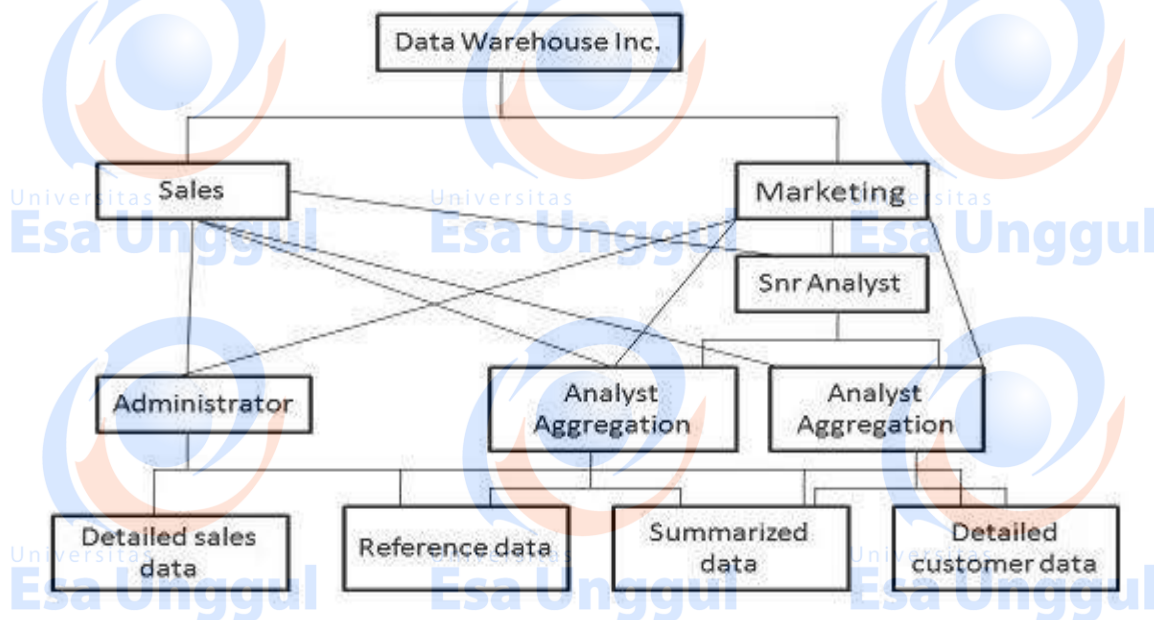


But if each department accesses different data, then we should design the security access for each department separately. This can be achieved by departmental data marts. Since these data marts are separated from the data warehouse, we can enforce separate security restrictions on each data mart. This approach is shown in the following figure.



Classification Based on Role

If the data is generally available to all the departments, then it is useful to follow the role access hierarchy. In other words, if the data is generally accessed by all the departments, then apply security restrictions as per the role of the user. The role access hierarchy is shown in the following figure.



Audit Requirements

Auditing is a subset of security, a costly activity. Auditing can cause heavy overheads on the system. To complete an audit in time, we require more hardware and therefore, it is recommended that wherever possible, auditing should be switched off. Audit requirements can be categorized as follows:

- Connections
- Disconnections

- Data access
- Data change

Note: For each of the above-mentioned categories, it is necessary to audit success, failure, or both. From the perspective of security reasons, the auditing of failures are very important. Auditing of failure is important because they can highlight unauthorized or fraudulent access.

Network Requirements

Network security is as important as other securities. We cannot ignore the network security requirement. We need to consider the following issues:

- Is it necessary to encrypt data before transferring it to the data warehouse?
- Are there restrictions on which network routes the data can take?

These restrictions need to be considered carefully. Following are the points to remember:

- The process of encryption and decryption will increase overheads. It would require more processing power and processing time.
- The cost of encryption can be high if the system is already a loaded system because the encryption is borne by the source system.

Data Movement

There exist potential security implications while moving the data. Suppose we need to transfer some restricted data as a flat file to be loaded. When the data is loaded into the data warehouse, the following questions are raised:

- Where is the flat file stored?
- Who has access to that disk space?

If we talk about the backup of these flat files, the following questions are raised:

- Do you backup encrypted or decrypted versions?
- Do these backups need to be made to special tapes that are stored separately?
- Who has access to these tapes?

Some other forms of data movement like query result sets also need to be considered. The questions raised while creating the temporary table are as follows:

- Where is that temporary table to be held?
- How do you make such table visible?

We should avoid the accidental flouting of security restrictions. If a user with access to the restricted data can generate accessible temporary tables, data can be visible to non-authorized users. We can overcome this problem by having a separate temporary area for users with access to restricted data.

Documentation

The audit and security requirements need to be properly documented. This will be treated as a part of justification. This document can contain all the information gathered from:

- Data classification
- User classification
- Network requirements
- Data movement and storage requirements
- All auditable actions

Impact of Security on Design

Security affects the application code and the development timescales. Security affects the following area:

- Application development
- Database design
- Testing

Application Development

Security affects the overall application development and it also affects the design of the important components of the data warehouse such as load manager, warehouse manager, and query manager. The load manager may require checking code to filter record and place them in different locations. More transformation rules may also be required to hide certain data. Also there may be requirements of extra metadata to handle any extra objects.

To create and maintain extra views, the warehouse manager may require extra codes to enforce security. Extra checks may have to be coded into the data warehouse to prevent it from being fooled into moving data into a location where it should not be available. The query manager requires the changes to handle any access restrictions. The query manager will need to be aware of all extra views and aggregations.

Database Design

The database layout is also affected because when security measures are implemented, there is an increase in the number of views and tables. Adding security increases the size of the database and hence increases the complexity of the database design and management. It will also add complexity to the backup management and recovery plan.

Testing

Testing the data warehouse is a complex and lengthy process. Adding security to the data warehouse also affects the testing time complexity. It affects the testing in the following two ways:

- It will increase the time required for integration and system testing.
- There is added functionality to be tested which will increase the size of the testing suite.

DDBMS - SECURITY IN DISTRIBUTED DATABASES

https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_security_distributed_databases.htm

Copyright © tutorialspoint.com

A distributed system needs additional security measures than centralized system, since there are many users, diversified data, multiple sites and distributed control. In this chapter, we will look into the various facets of distributed database security.

In distributed communication systems, there are two types of intruders –

- **Passive eavesdroppers** – They monitor the messages and get hold of private information.
- **Active attackers** – They not only monitor the messages but also corrupt data by inserting new data or modifying existing data.

Security measures encompass security in communications, security in data and data auditing.

Communications Security

In a distributed database, a lot of data communication takes place owing to the diversified location of data, users and transactions. So, it demands secure communication between users and databases and between the different database environments.

Security in communication encompasses the following –

- Data should not be corrupt during transfer.
- The communication channel should be protected against both passive eavesdroppers and active attackers.
- In order to achieve the above stated requirements, well-defined security algorithms and protocols should be adopted.

Two popular, consistent technologies for achieving end-to-end secure communications are –

- Secure Socket Layer Protocol or Transport Layer Security Protocol.
- Virtual Private Networks *VPN*.

Data Security

In distributed systems, it is imperative to adopt measure to secure data apart from communications. The data security measures are –

- **Authentication and authorization** – These are the access control measures adopted to ensure that only authentic users can use the database. To provide authentication digital certificates are used. Besides, login is restricted through username/password combination.
- **Data encryption** – The two approaches for data encryption in distributed systems are –
 - Internal to distributed database approach: The user applications encrypt the data and then store the encrypted data in the database. For using the stored data, the applications fetch the encrypted data from the database and then decrypt it.
 - External to distributed database: The distributed database system has its own encryption capabilities. The user applications store data and retrieve them without realizing that the data is stored in an encrypted form in the database.
- **Validated input** – In this security measure, the user application checks for each input before it can be used for updating the database. An un-validated input can cause a wide range of exploits like buffer overrun, command injection, cross-site scripting and corruption in data.

Data Auditing

A database security system needs to detect and monitor security violations, in order to ascertain the security measures it should adopt. It is often very difficult to detect breach of security at the time of occurrences. One method to identify security violations is to examine audit logs. Audit logs contain information such as –

- Date, time and site of failed access attempts.
- Details of successful access attempts.
- Vital modifications in the database system.
- Access of huge amounts of data, particularly from databases in multiple sites.

All the above information gives an insight of the activities in the database. A periodical analysis of the log helps to identify any unnatural activity along with its site and time of occurrence. This log is ideally stored in a separate server so that it is inaccessible to attackers.



Modul 12

Distributed-Object-System



2. OOAD – Object Model

The object model visualizes the elements in a software application in terms of objects. In this chapter, we will look into the basic concepts and terminologies of object-oriented systems.

Objects and Classes

The concepts of objects and classes are intrinsically linked with each other and form the foundation of object-oriented paradigm.

Object

An object is a real-world element in an object-oriented environment that may have a physical or a conceptual existence. Each object has:

- Identity that distinguishes it from other objects in the system.
- State that determines the characteristic properties of an object as well as the values of the properties that the object holds.
- Behavior that represents externally visible activities performed by an object in terms of changes in its state.

Objects can be modelled according to the needs of the application. An object may have a physical existence, like a customer, a car, etc.; or an intangible conceptual existence, like a project, a process, etc.

Class

A class represents a collection of objects having same characteristic properties that exhibit common behavior. It gives the blueprint or description of the objects that can be created from it. Creation of an object as a member of a class is called instantiation. Thus, object is an instance of a class.

The constituents of a class are:

- A set of attributes for the objects that are to be instantiated from the class. Generally, different objects of a class have some difference in the values of the attributes. Attributes are often referred as class data.
- A set of operations that portray the behavior of the objects of the class. Operations are also referred as functions or methods.

Example

Let us consider a simple class, Circle, that represents the geometrical figure circle in a two-dimensional space. The attributes of this class can be identified as follows:

- x-coord, to denote x-coordinate of the center
- y-coord, to denote y-coordinate of the center
- a, to denote the radius of the circle

Some of its operations can be defined as follows:

- `findArea()`, method to calculate area
- `findCircumference()`, method to calculate circumference
- `scale()`, method to increase or decrease the radius

During instantiation, values are assigned for at least some of the attributes. If we create an object `my_circle`, we can assign values like `x-coord : 2`, `y-coord : 3`, and `a : 4` to depict its state. Now, if the operation `scale()` is performed on `my_circle` with a scaling factor of 2, the value of the variable `a` will become 8. This operation brings a change in the state of `my_circle`, i.e., the object has exhibited certain behavior.

Encapsulation and Data Hiding

Encapsulation

Encapsulation is the process of binding both attributes and methods together within a class. Through encapsulation, the internal details of a class can be hidden from outside. It permits the elements of the class to be accessed from outside only through the interface provided by the class.

Data Hiding

Typically, a class is designed such that its data (attributes) can be accessed only by its class methods and insulated from direct outside access. This process of insulating an object's data is called data hiding or information hiding.

Example

In the class `Circle`, data hiding can be incorporated by making attributes invisible from outside the class and adding two more methods to the class for accessing class data, namely:

- `setValues()`, method to assign values to `x-coord`, `y-coord`, and `a`
- `getValues()`, method to retrieve values of `x-coord`, `y-coord`, and `a`

Here the private data of the object `my_circle` cannot be accessed directly by any method that is not encapsulated within the class `Circle`. It should instead be accessed through the methods `setValues()` and `getValues()`.

Message Passing

Any application requires a number of objects interacting in a harmonious manner. Objects in a system may communicate with each other using message passing. Suppose a system has two objects: `obj1` and `obj2`. The object `obj1` sends a message to object `obj2`, if `obj1` wants `obj2` to execute one of its methods.

The features of message passing are:

- Message passing between two objects is generally unidirectional.
- Message passing enables all interactions between objects.
- Message passing essentially involves invoking class methods.

- Objects in different processes can be involved in message passing.

Inheritance

Inheritance is the mechanism that permits new classes to be created out of existing classes by extending and refining its capabilities. The existing classes are called the base classes/parent classes/super-classes, and the new classes are called the derived classes/child classes/subclasses. The subclass can inherit or derive the attributes and methods of the super-class(es) provided that the super-class allows so. Besides, the subclass may add its own attributes and methods and may modify any of the super-class methods. Inheritance defines an "is – a" relationship.

Example

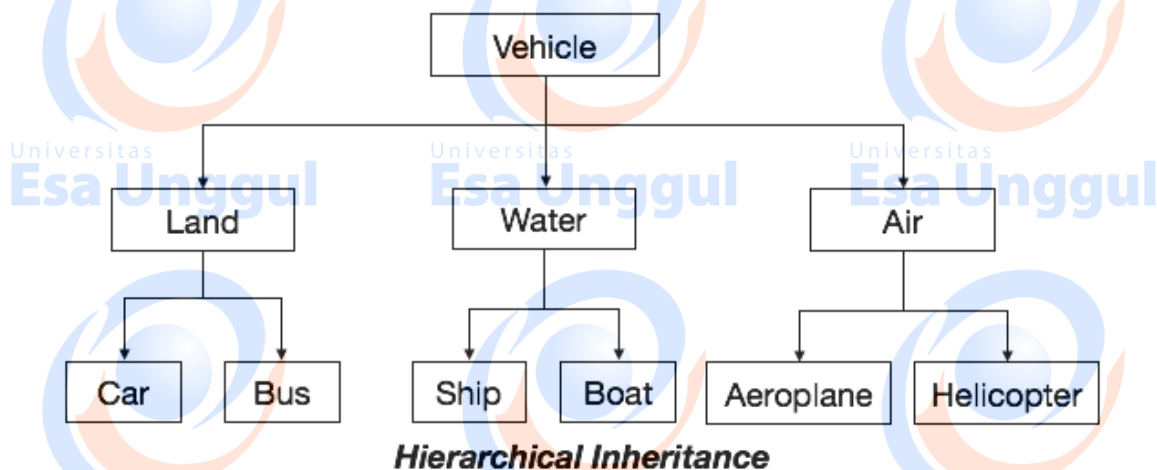
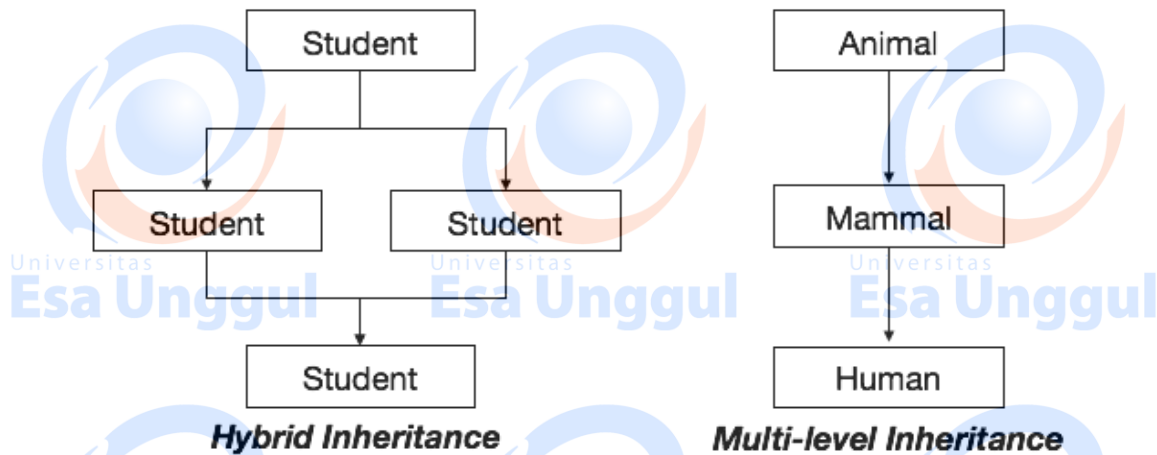
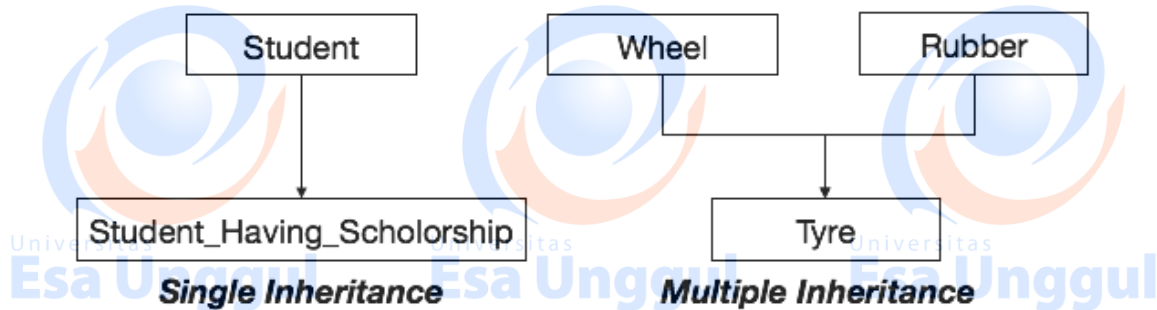
From a class Mammal, a number of classes can be derived such as Human, Cat, Dog, Cow, etc. Humans, cats, dogs, and cows all have the distinct characteristics of mammals. In addition, each has its own particular characteristics. It can be said that a cow "is – a" mammal.

Types of Inheritance

- **Single Inheritance** : A subclass derives from a single super-class.
- **Multiple Inheritance** : A subclass derives from more than one super-classes.
- **Multilevel Inheritance** : A subclass derives from a super-class which in turn is derived from another class and so on.
- **Hierarchical Inheritance** : A class has a number of subclasses each of which may have subsequent subclasses, continuing for a number of levels, so as to form a tree structure.
- **Hybrid Inheritance** : A combination of multiple and multilevel inheritance so as to form a lattice structure.

The following figure depicts the examples of different types of inheritance.





Polymorphism

Polymorphism is originally a Greek word that means the ability to take multiple forms. In object-oriented paradigm, polymorphism implies using operations in different ways, depending upon the instance they are operating upon. Polymorphism allows objects with different internal structures to have a common external interface. Polymorphism is particularly effective while implementing inheritance.

Example



Let us consider two classes, Circle and Square, each with a method findArea(). Though the name and purpose of the methods in the classes are same, the internal implementation, i.e., the procedure of calculating area is different for each class. When an object of class Circle invokes its findArea() method, the operation finds the area of the circle without any conflict with the findArea() method of the Square class.

Generalization and Specialization

Generalization and specialization represent a hierarchy of relationships between classes, where subclasses inherit from super-classes.

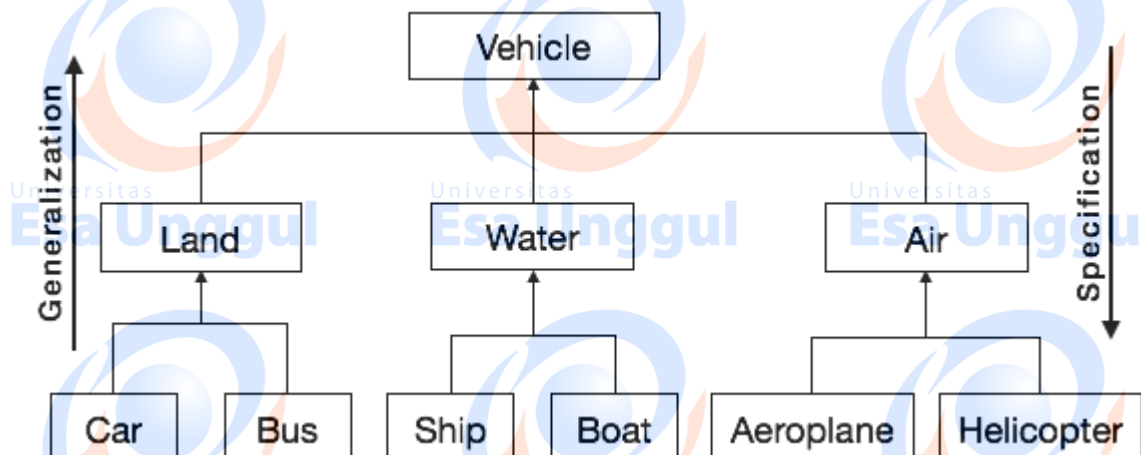
Generalization

In the generalization process, the common characteristics of classes are combined to form a class in a higher level of hierarchy, i.e., subclasses are combined to form a generalized super-class. It represents an "is - a - kind - of" relationship. For example, "car is a kind of land vehicle", or "ship is a kind of water vehicle".

Specialization

Specialization is the reverse process of generalization. Here, the distinguishing features of groups of objects are used to form specialized classes from existing classes. It can be said that the subclasses are the specialized versions of the super-class.

The following figure shows an example of generalization and specialization.



Links and Association

Link

A link represents a connection through which an object collaborates with other objects. Rumbaugh has defined it as "a physical or conceptual connection between objects". Through a link, one object may invoke the methods or navigate through another object. A link depicts the relationship between two or more objects.

Association

Association is a group of links having common structure and common behavior. Association depicts the relationship between objects of one or more classes. A link can be defined as an instance of an association.

Degree of an Association

Degree of an association denotes the number of classes involved in a connection. Degree may be unary, binary, or ternary.

- A **unary relationship** connects objects of the same class.
- A **binary relationship** connects objects of two classes.
- A **ternary relationship** connects objects of three or more classes.

Cardinality Ratios of Associations

Cardinality of a binary association denotes the number of instances participating in an association. There are three types of cardinality ratios, namely:

- **One-to-One** : A single object of class A is associated with a single object of class B.
- **One-to-Many** : A single object of class A is associated with many objects of class B.
- **Many-to-Many** : An object of class A may be associated with many objects of class B and conversely an object of class B may be associated with many objects of class A.

Aggregation or Composition

Aggregation or composition is a relationship among classes by which a class can be made up of any combination of objects of other classes. It allows objects to be placed directly within the body of other classes. Aggregation is referred as a "part-of" or "has-a" relationship, with the ability to navigate from the whole to its parts. An aggregate object is an object that is composed of one or more other objects.

Example

In the relationship, "a car has-a motor", car is the whole object or the aggregate, and the motor is a "part-of" the car. Aggregation may denote:

- **Physical containment** : Example, a computer is composed of monitor, CPU, mouse, keyboard, and so on.
- **Conceptual containment** : Example, shareholder has-a share.

Benefits of Object Model

Now that we have gone through the core concepts pertaining to object orientation, it would be worthwhile to note the advantages that this model has to offer.

The benefits of using the object model are:

- It helps in faster development of software.

- It is easy to maintain. Suppose a module develops an error, then a programmer can fix that particular module, while the other parts of the software are still up and running.
- It supports relatively hassle-free upgrades.
- It enables reuse of objects, designs, and functions.
- It reduces development risks, particularly in integration of complex systems.



3. OOAD – Object-Oriented System

We know that the Object-Oriented Modelling (OOM) technique visualizes things in an application by using models organized around objects. Any software development approach goes through the following stages:

- Analysis,
- Design, and
- Implementation.

In object-oriented software engineering, the software developer identifies and organizes the application in terms of object-oriented concepts, prior to their final representation in any specific programming language or software tools.

Phases in Object-Oriented Software Development

The major phases of software development using object-oriented methodology are object-oriented analysis, object-oriented design, and object-oriented implementation.

Object-Oriented Analysis

In this stage, the problem is formulated, user requirements are identified, and then a model is built based upon real-world objects. The analysis produces models on how the desired system should function and how it must be developed. The models do not include any implementation details so that it can be understood and examined by any non-technical application expert.

Object-Oriented Design

Object-oriented design includes two main stages, namely, system design and object design.

System Design

In this stage, the complete architecture of the desired system is designed. The system is conceived as a set of interacting subsystems that in turn is composed of a hierarchy of interacting objects, grouped into classes. System design is done according to both the system analysis model and the proposed system architecture. Here, the emphasis is on the objects comprising the system rather than the processes in the system.

Object Design

In this phase, a design model is developed based on both the models developed in the system analysis phase and the architecture designed in the system design phase. All the classes required are identified. The designer decides whether:

- new classes are to be created from scratch,
- any existing classes can be used in their original form, or
- new classes should be inherited from the existing classes.

The associations between the identified classes are established and the hierarchies of classes are identified. Besides, the developer designs the internal details of the classes and their associations, i.e., the data structure for each attribute and the algorithms for the operations.

Object-Oriented Implementation and Testing

In this stage, the design model developed in the object design is translated into code in an appropriate programming language or software tool. The databases are created and the specific hardware requirements are ascertained. Once the code is in shape, it is tested using specialized techniques to identify and remove the errors in the code.





Modul 13

Distributed File System



HIVE - INTRODUCTION

http://www.tutorialspoint.com/hive/hive_introduction.htm

Copyright © tutorialspoint.com

The term 'Big Data' is used for collections of large datasets that include huge volume, high velocity, and a variety of data that is increasing day by day. Using traditional data management systems, it is difficult to process Big Data. Therefore, the Apache Software Foundation introduced a framework called Hadoop to solve Big Data management and processing challenges.

Hadoop

Hadoop is an open-source framework to store and process Big Data in a distributed environment. It contains two modules, one is MapReduce and another is Hadoop Distributed File System *HDFS*.

- **MapReduce:** It is a parallel programming model for processing large amounts of structured, semi-structured, and unstructured data on large clusters of commodity hardware.
- **HDFS:** Hadoop Distributed File System is a part of Hadoop framework, used to store and process the datasets. It provides a fault-tolerant file system to run on commodity hardware.

The Hadoop ecosystem contains different sub-projects *tools* such as Sqoop, Pig, and Hive that are used to help Hadoop modules.

- **Sqoop:** It is used to import and export data to and from between HDFS and RDBMS.
- **Pig:** It is a procedural language platform used to develop a script for MapReduce operations.
- **Hive:** It is a platform used to develop SQL type scripts to do MapReduce operations.

Note: There are various ways to execute MapReduce operations:

- The traditional approach using Java MapReduce program for structured, semi-structured, and unstructured data.
- The scripting approach for MapReduce to process structured and semi structured data using Pig.
- The Hive Query Language *HiveQL* or *HQL* for MapReduce to process structured data using Hive.

What is Hive

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

Hive is not

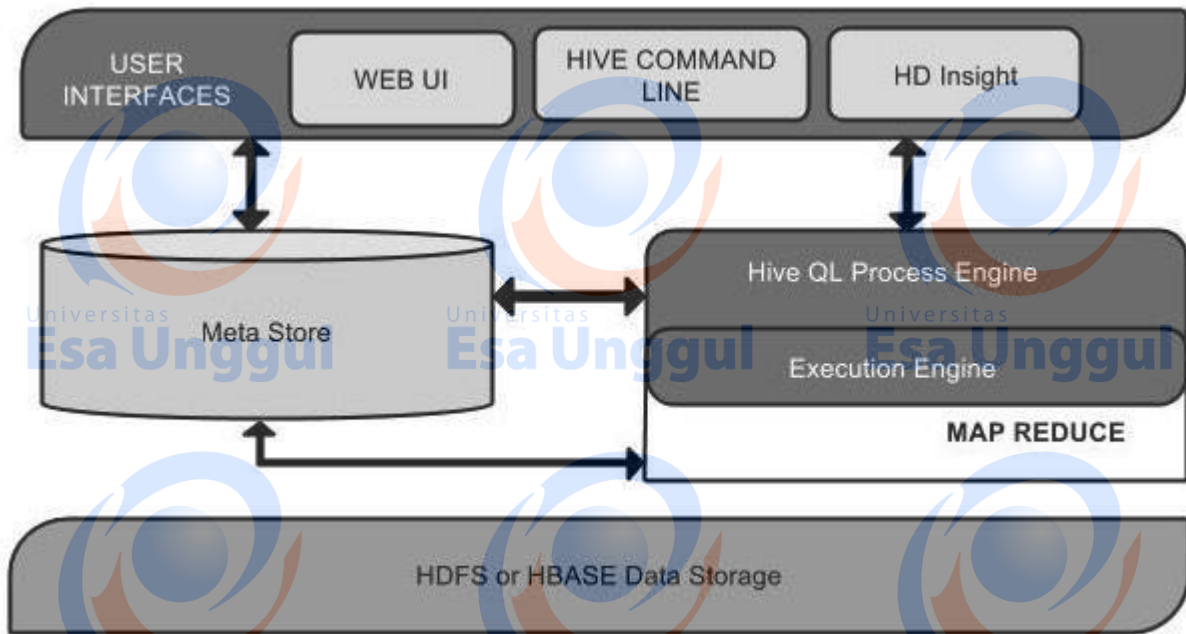
- A relational database
- A design for OnLine Transaction Processing *OLTP*
- A language for real-time queries and row-level updates

Features of Hive

- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.

Architecture of Hive

The following component diagram depicts the architecture of Hive:

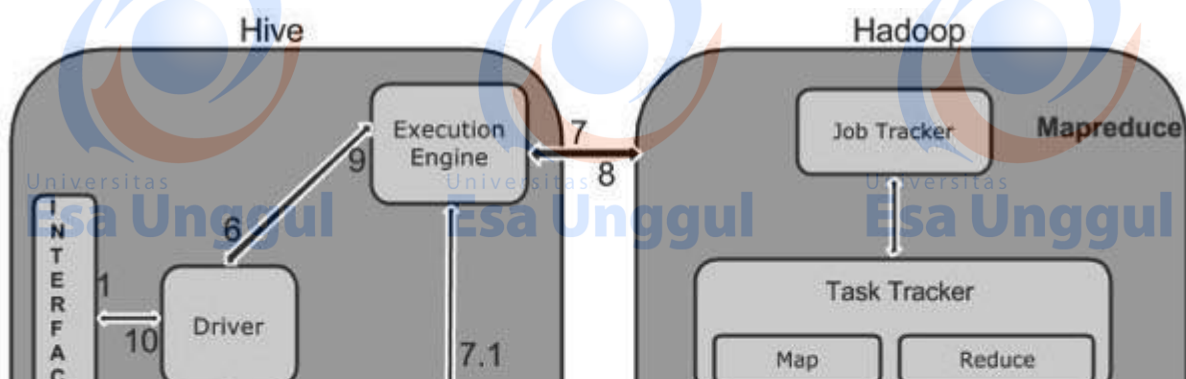


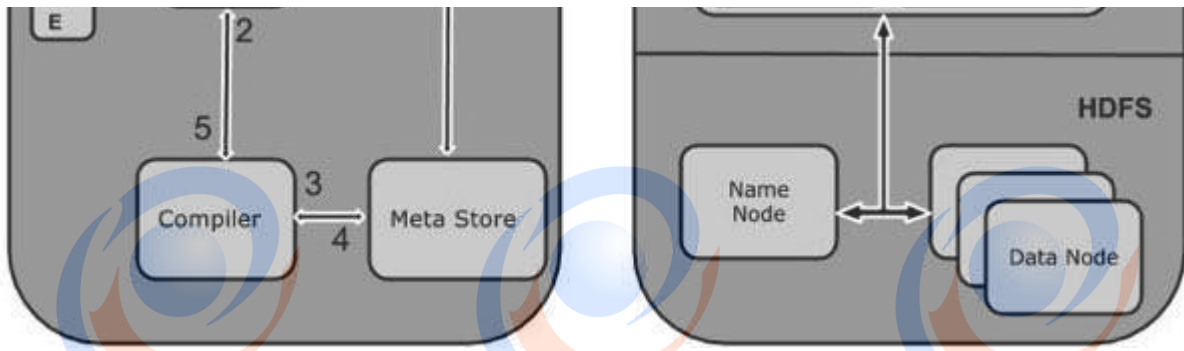
This component diagram contains different units. The following table describes each unit:

Unit Name	Operation
User Interface	Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight <i>InWindowsserver</i> .
Meta Store	Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.
HiveQL Process Engine	HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.
Execution Engine	The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of MapReduce.
HDFS or HBASE	Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

Working of Hive

The following diagram depicts the workflow between Hive and Hadoop.





The following table defines how Hive interacts with Hadoop framework:

Step No.	Operation
1	<p>Execute Query</p> <p>The Hive interface such as Command Line or Web UI sends query to Driver <i>anydatabase drivers such as JDBC, ODBC, etc.</i> to execute.</p>
2	<p>Get Plan</p> <p>The driver takes the help of query compiler that parses the query to check the syntax and query plan or the requirement of query.</p>
3	<p>Get Metadata</p> <p>The compiler sends metadata request to Metastore <i>anydatabase</i>.</p>
4	<p>Send Metadata</p> <p>Metastore sends metadata as a response to the compiler.</p>
5	<p>Send Plan</p> <p>The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete.</p>
6	<p>Execute Plan</p> <p>The driver sends the execute plan to the execution engine.</p>
7	<p>Execute Job</p> <p>Internally, the process of execution job is a MapReduce job. The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Here, the query executes MapReduce job.</p>
7.1	<p>Metadata Ops</p> <p>Meanwhile in execution, the execution engine can execute metadata operations with Metastore.</p>
8	<p>Fetch Result</p> <p>The execution engine receives the results from Data nodes.</p>
9	<p>Send Results</p>

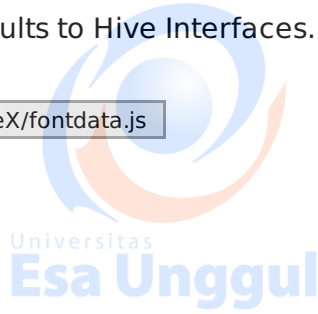
The execution engine sends those resultant values to the driver.

10

Send Results

The driver sends the results to Hive Interfaces.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js



MAPREDUCE - ALGORITHM

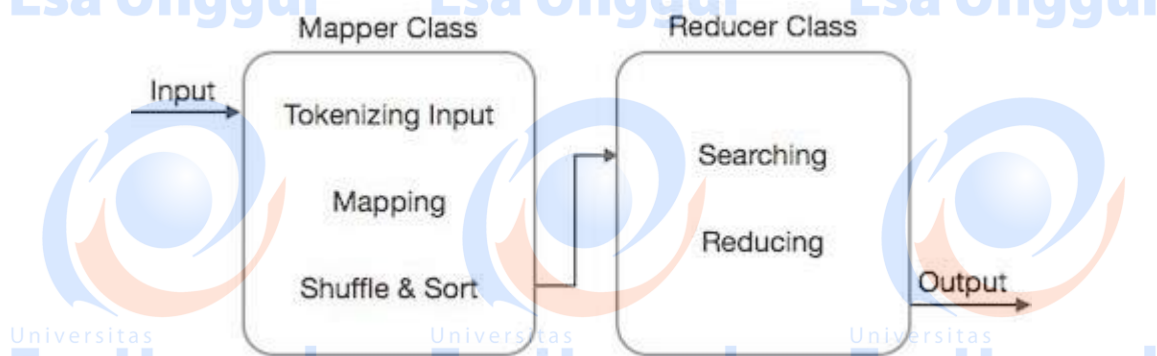
http://www.tutorialspoint.com/map_reduce/map_reduce_algorithm.htm

Copyright © tutorialspoint.com

The MapReduce algorithm contains two important tasks, namely Map and Reduce.

- The map task is done by means of Mapper Class
- The reduce task is done by means of Reducer Class.

Mapper class takes the input, tokenizes it, maps and sorts it. The output of Mapper class is used as input by Reducer class, which in turn searches matching pairs and reduces them.



MapReduce implements various mathematical algorithms to divide a task into small parts and assign them to multiple systems. In technical terms, MapReduce algorithm helps in sending the Map & Reduce tasks to appropriate servers in a cluster.

These mathematical algorithms may include the following –

- Sorting
- Searching
- Indexing
- TF-IDF

Sorting

Sorting is one of the basic MapReduce algorithms to process and analyze data. MapReduce implements sorting algorithm to automatically sort the output key-value pairs from the mapper by their keys.

- Sorting methods are implemented in the mapper class itself.
- In the Shuffle and Sort phase, after tokenizing the values in the mapper class, the **Context** class *user-definedclass* collects the matching valued keys as a collection.
- To collect similar key-value pairs *intermediatekeys*, the Mapper class takes the help of **RawComparator** class to sort the key-value pairs.
- The set of intermediate key-value pairs for a given Reducer is automatically sorted by Hadoop to form key-values $K2, V2, V2, \dots$ before they are presented to the Reducer.

Searching

Searching plays an important role in MapReduce algorithm. It helps in the combiner phase *optional* and in the Reducer phase. Let us try to understand how Searching works with the help of an example.

Example

The following example shows how MapReduce employs Searching algorithm to find out the details of the employee who draws the highest salary in a given employee dataset.

- Let us assume we have employee data in four different files – A, B, C, and D. Let us also assume there are duplicate employee records in all four files because of importing the employee data from all database tables repeatedly. See the following illustration.

name, salary	name, salary	name, salary	name, salary
satish, 26000	gopal, 50000	satish, 26000	satish, 26000
Krishna, 25000	Krishna, 25000	kiran, 45000	Krishna, 25000
Satishk, 15000	Satishk, 15000	Satishk, 15000	manisha, 45000
Raju, 10000	Raju, 10000	Raju, 10000	Raju, 10000

- The Map phase** processes each input file and provides the employee data in key-value pairs $\langle k, v \rangle$: $\langle empname, salary \rangle$. See the following illustration.

$\langle satish, 26000 \rangle$	$\langle gopal, 50000 \rangle$	$\langle satish, 26000 \rangle$	$\langle satish, 26000 \rangle$
$\langle Krishna, 25000 \rangle$	$\langle Krishna, 25000 \rangle$	$\langle kiran, 45000 \rangle$	$\langle Krishna, 25000 \rangle$
$\langle Satishk, 15000 \rangle$	$\langle Satishk, 15000 \rangle$	$\langle Satishk, 15000 \rangle$	$\langle manisha, 45000 \rangle$
$\langle Raju, 10000 \rangle$	$\langle Raju, 10000 \rangle$	$\langle Raju, 10000 \rangle$	$\langle Raju, 10000 \rangle$

- The combiner phase** *searching technique* will accept the input from the Map phase as a key-value pair with employee name and salary. Using searching technique, the combiner will check all the employee salary to find the highest salaried employee in each file. See the following snippet.

```

<k: employee name, v: salary>
Max= the salary of an first employee. Treated as max salary
if(v(second employee).salary > Max){
    Max = v(salary);
}
else{
    Continue checking;
}

```

The expected result is as follows –

$\langle satish, 26000 \rangle$	$\langle gopal, 50000 \rangle$	$\langle kiran, 45000 \rangle$	$\langle manisha, 45000 \rangle$
---------------------------------	--------------------------------	--------------------------------	----------------------------------

- Reducer phase** – Form each file, you will find the highest salaried employee. To avoid redundancy, check all the $\langle k, v \rangle$ pairs and eliminate duplicate entries, if any. The same algorithm is used in between the four $\langle k, v \rangle$ pairs, which are coming from four input files.

The final output should be as follows –

```
<gopal, 50000>
```

Indexing

Normally indexing is used to point to a particular data and its address. It performs batch indexing on the input files for a particular Mapper.

The indexing technique that is normally used in MapReduce is known as **inverted index**. Search engines like Google and Bing use inverted indexing technique. Let us try to understand how Indexing works with the help of a simple example.

Example

The following text is the input for inverted indexing. Here T[0], T[1], and T[2] are the file names and their content are in double quotes.

```
T[0] = "it is what it is"  
T[1] = "what is it"  
T[2] = "it is a banana"
```

After applying the Indexing algorithm, we get the following output –

```
"a": {2}  
"banana": {2}  
"is": {0, 1, 2}  
"it": {0, 1, 2}  
"what": {0, 1}
```

Here "a": {2} implies the term "a" appears in the T[2] file. Similarly, "is": {0, 1, 2} implies the term "is" appears in the files T[0], T[1], and T[2].

TF-IDF

TF-IDF is a text processing algorithm which is short for Term Frequency – Inverse Document Frequency. It is one of the common web analysis algorithms. Here, the term 'frequency' refers to the number of times a term appears in a document.

Term Frequency *TF*

It measures how frequently a particular term occurs in a document. It is calculated by the number of times a word appears in a document divided by the total number of words in that document.

```
TF(the) = (Number of times term the 'the' appears in a document) / (Total number of terms in the document)
```

Inverse Document Frequency *IDF*

It measures the importance of a term. It is calculated by the number of documents in the text database divided by the number of documents where a specific term appears.

While computing TF, all the terms are considered equally important. That means, TF counts the term frequency for normal words like "is", "a", "what", etc. Thus we need to know the frequent terms while scaling up the rare ones, by computing the following –

```
IDF(the) = log_e(Total number of documents / Number of documents with term 'the' in it).
```

The algorithm is explained below with the help of a small example.

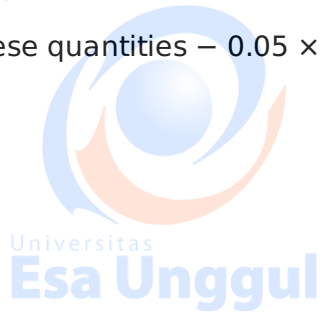
Example

Consider a document containing 1000 words, wherein the word **hive** appears 50 times. The TF for **hive** is then $50/1000 = 0.05$.

Now, assume we have 10 million documents and the word **hive** appears in 1000 of these. Then, the IDF is calculated as $\log_{10}(10,000,000/1,000) = 4$.

The TF-IDF weight is the product of these quantities – $0.05 \times 4 = 0.20$.

Loading [Mathjax]/jax/output/HTML-CSS/jax.js



MAPREDUCE - HADOOP ADMINISTRATION

http://www.tutorialspoint.com/map_reduce/map_reduce_hadoop_administration.htm Copyright © tutorialspoint.com

This chapter explains Hadoop administration which includes both HDFS and MapReduce administration.

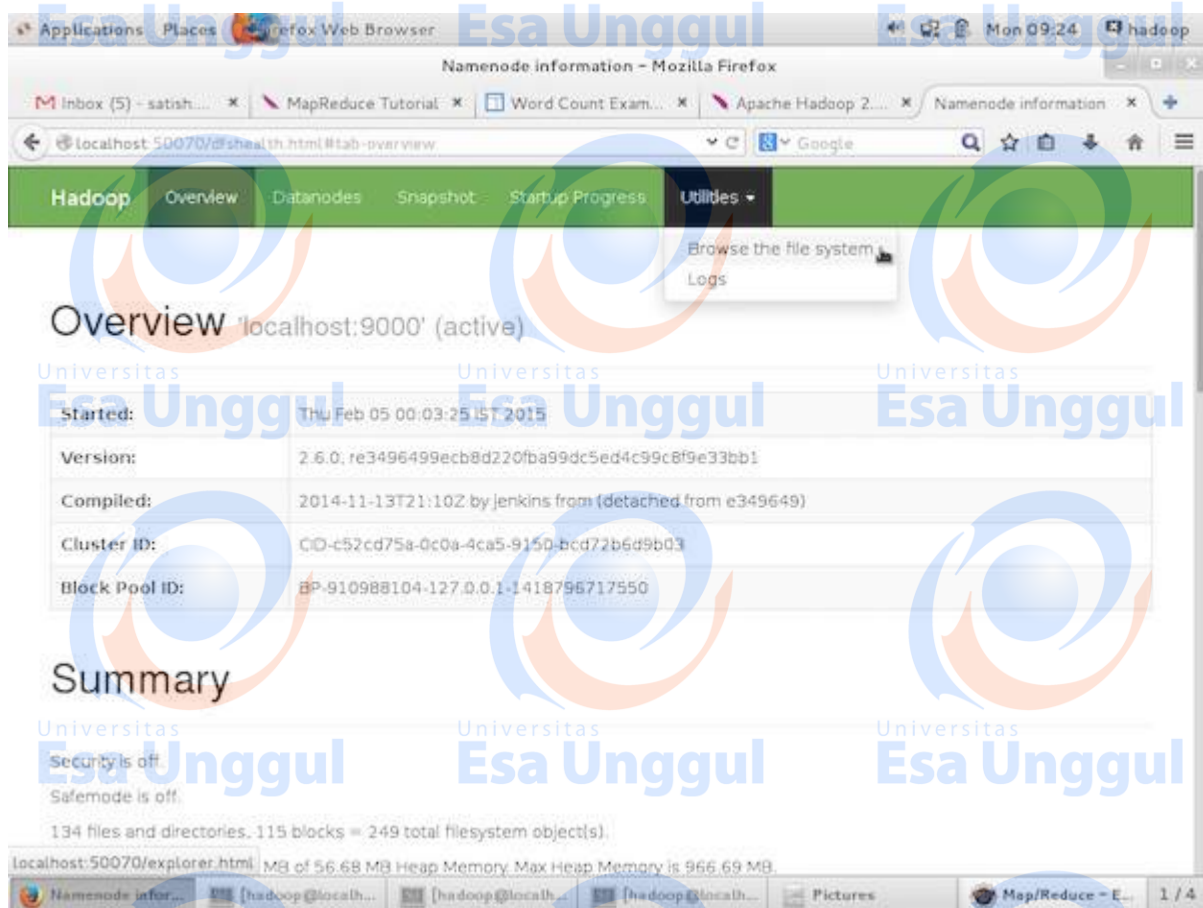
- HDFS administration includes monitoring the HDFS file structure, locations, and the updated files.
- MapReduce administration includes monitoring the list of applications, configuration of nodes, application status, etc.

HDFS Monitoring

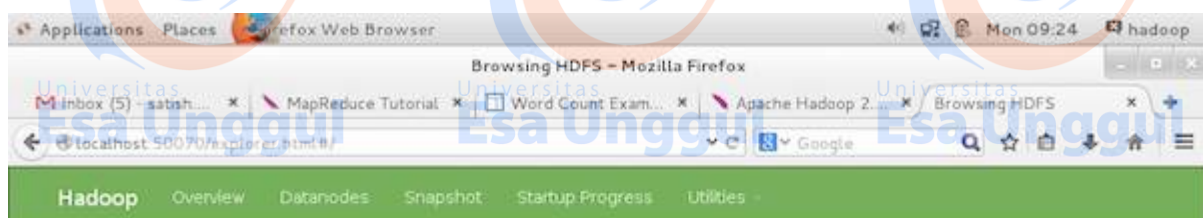
HDFS *HadoopDistributedFileSystem* contains the user directories, input files, and output files. Use the MapReduce commands, **put** and **get**, for storing and retrieving.

After starting the Hadoop framework *daemons* by passing the command “start-all.sh” on “/\$HADOOP_HOME/sbin”, pass the following URL to the browser “http://localhost:50070”. You should see the following screen on your browser.

The following screenshot shows how to browse the browse HDFS.



The following screenshot show the file structure of HDFS. It shows the files in the “/user/hadoop” directory.

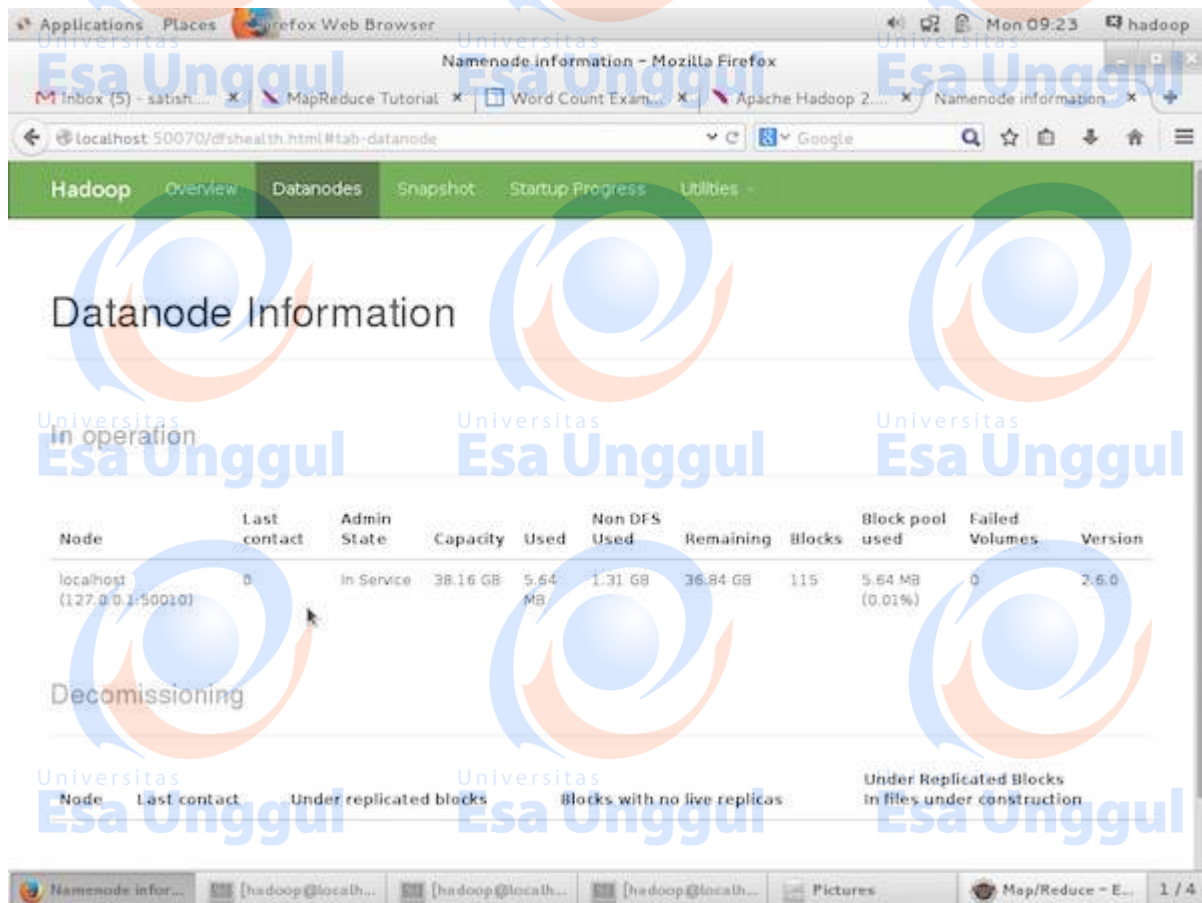


Browse Directory



Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	hadoop	supergroup	0 B	0	0 B	querieset
drwx-----	hadoop	supergroup	0 B	0	0 B	tmp
drwxr-xr-x	hadoop	supergroup	0 B	0	0 B	user

The following screenshot shows the Datanode information in a cluster. Here you can find one node with its configurations and capacities.



Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
localhost (127.0.0.1:50010)	0	In Service	38.16 GB	5.64 MB	1.31 GB	36.84 GB	115	5.64 MB (0.01%)	0	2.6.0

Decommissioning

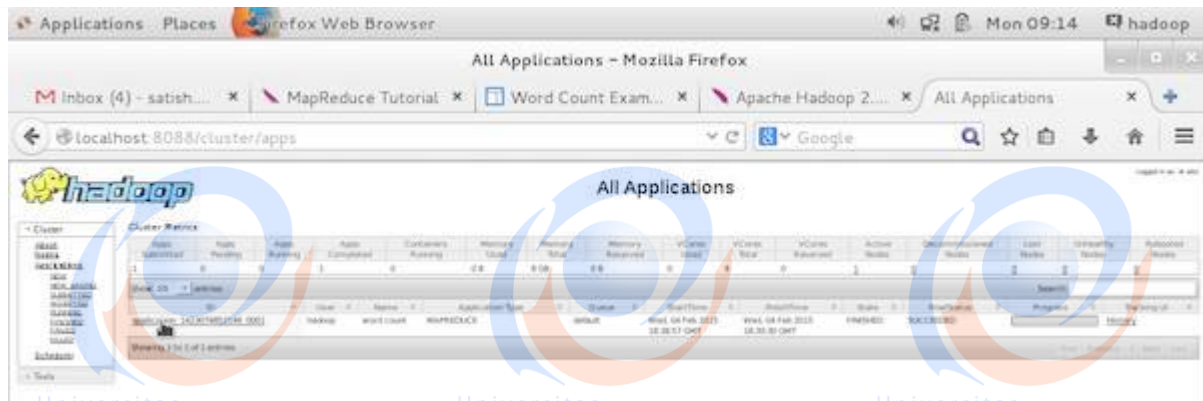
Node Last contact Under replicated blocks Blocks with no live replicas Under Replicated Blocks in files under construction

MapReduce Job Monitoring

A MapReduce application is a collection of jobs *Mapjob*, *Combiner*, *Partitioner*, and *Reducejob*. It is mandatory to monitor and maintain the following –

- Configuration of datanode where the application is suitable.
- The number of datanodes and resources used per application.

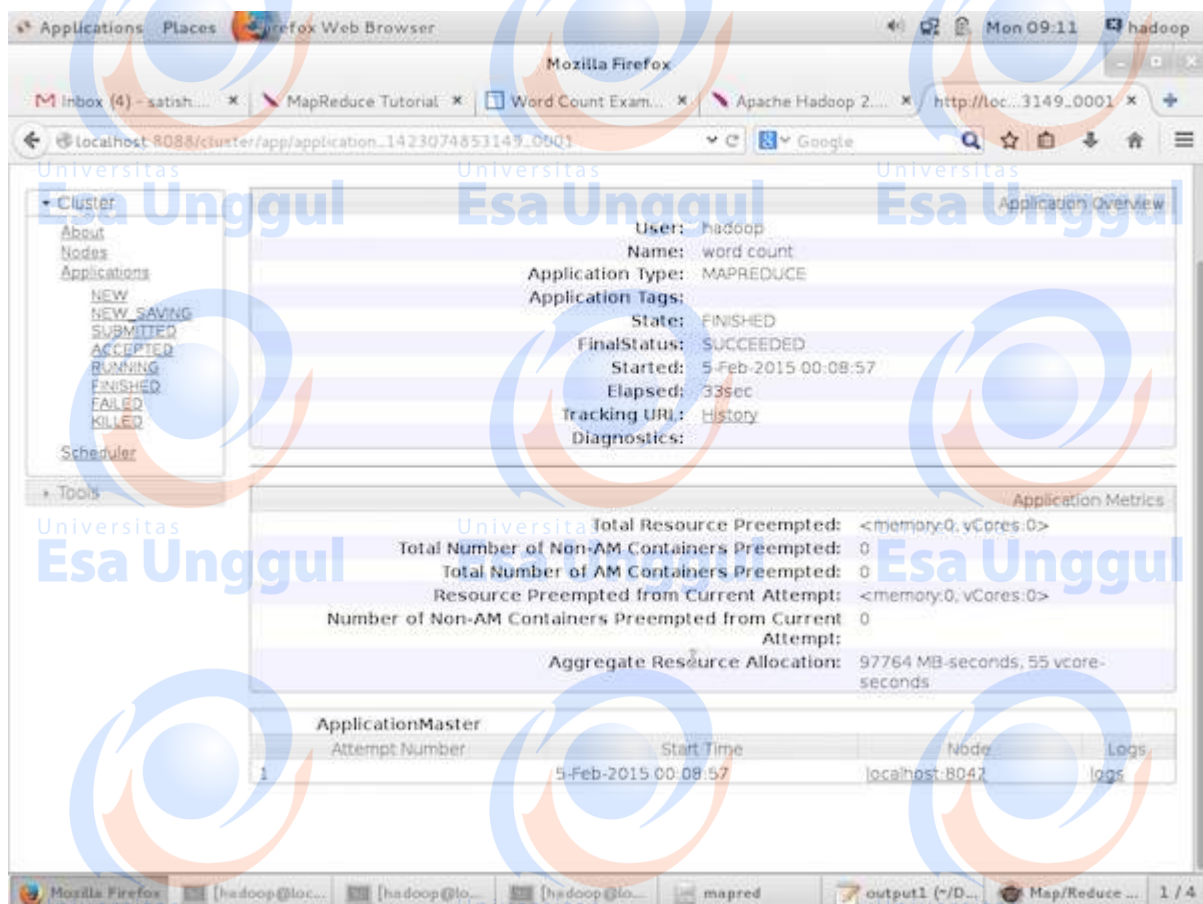
To monitor all these things, it is imperative that we should have a user interface. After starting the Hadoop framework by passing the command “start-all.sh” on “/\$HADOOP_HOME/sbin”, pass the following URL to the browser “http://localhost:8080”. You should see the following screen on your browser.



In the above screenshot, the hand pointer is on the application ID. Just click on it to find the following screen on your browser. It describes the following –

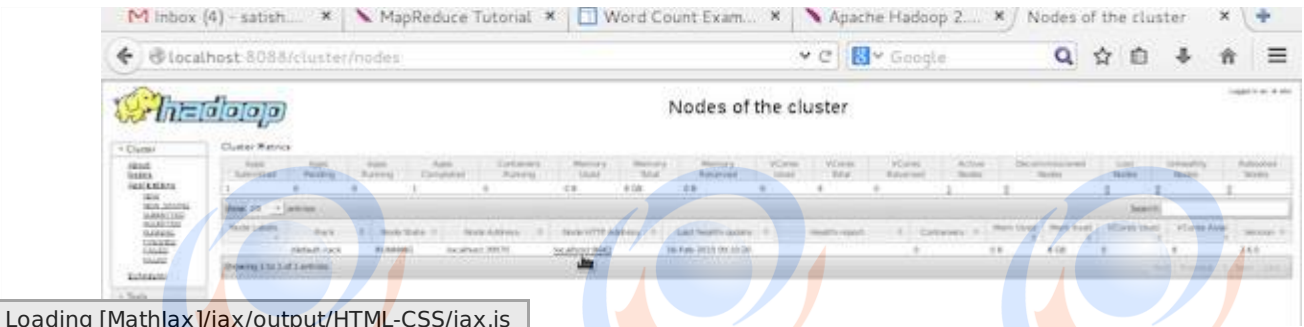
- On which user the current Web application is running
- The application name
- Type of that application
- Current status, Final status
- Application started time, elapsed *completedtime*, if it is complete at the time of monitoring
- The history of this application, i.e., log information
- And finally, the node information, i.e., the nodes that participated in running the application.

The following screenshot shows the details of a particular application –



The following screenshot describes the currently running nodes information. Here, the screenshot contains only one node. A hand pointer shows the localhost address of the running node.





Loading [Mathjax]/jax/output/HTML-CSS/jax.js

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

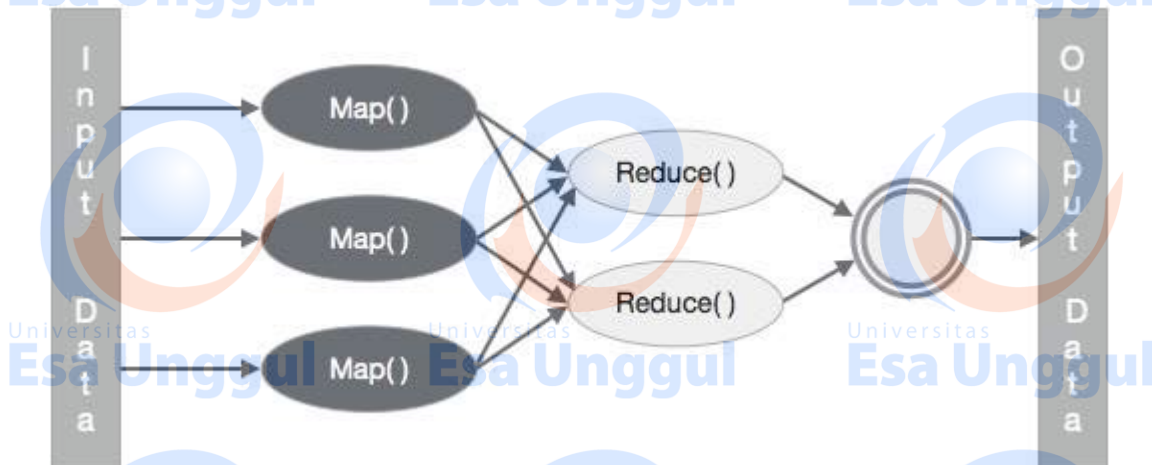
MAPREDUCE - HADOOP IMPLEMENTATION

MapReduce is a framework that is used for writing applications to process huge volumes of data on large clusters of commodity hardware in a reliable manner. This chapter takes you through the operation of MapReduce in Hadoop framework using Java.

MapReduce Algorithm

Generally MapReduce paradigm is based on sending map-reduce programs to computers where the actual data resides.

- During a MapReduce job, Hadoop sends Map and Reduce tasks to appropriate servers in the cluster.
- The framework manages all the details of data-passing like issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on the nodes with data on local disks that reduces the network traffic.
- After completing a given task, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.



Inputs and Outputs *Java Perspective*

The MapReduce framework operates on key-value pairs, that is, the framework views the input to the job as a set of key-value pairs and produces a set of key-value pair as the output of the job, conceivably of different types.

The key and value classes have to be serializable by the framework and hence, it is required to implement the Writable interface. Additionally, the key classes have to implement the WritableComparable interface to facilitate sorting by the framework.

Both the input and output format of a MapReduce job are in the form of key-value pairs –

Input <k1, v1> -> map -> <k2, v2>-> reduce -> <k3, v3> *Output*.

	Input	Output
Map	<k1, v1>	list < k2, v2 >
Reduce	<k2, listv2>	list < k3, v3 >

MapReduce Implementation

The following table shows the data regarding the electrical consumption of an organization. The table includes the monthly electrical consumption and the annual average for five consecutive years.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Avg
1979	23	23	2	43	24	25	26	26	26	26	25	26	25
1980	26	27	28	28	28	30	31	31	31	30	30	30	29
1981	31	32	32	32	33	34	35	36	36	34	34	34	34
1984	39	38	39	39	39	41	42	43	40	39	38	38	40
1985	38	39	39	39	39	41	41	41	00	40	39	39	45

We need to write applications to process the input data in the given table to find the year of maximum usage, the year of minimum usage, and so on. This task is easy for programmers with finite amount of records, as they will simply write the logic to produce the required output, and pass the data to the written application.

Let us now raise the scale of the input data. Assume we have to analyze the electrical consumption of all the large-scale industries of a particular state. When we write applications to process such bulk data,

- They will take a lot of time to execute.
- There will be heavy network traffic when we move data from the source to the network server.

To solve these problems, we have the MapReduce framework.

Input Data

The above data is saved as **sample.txt** and given as input. The input file looks as shown below.

```
1979 23 23 2 43 24 25 26 26 26 26 25 26 25
1980 26 27 28 28 28 30 31 31 31 30 30 30 29
1981 31 32 32 32 33 34 35 36 36 34 34 34 34
1984 39 38 39 39 39 41 42 43 40 39 38 38 40
1985 38 39 39 39 39 41 41 41 00 40 39 39 45
```

Example Program

The following program for the sample data uses MapReduce framework.

```
package hadoop;

import java.util.*;
import java.io.IOException;
import java.io.IOException;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;
```



```

public class ProcessUnits
{
    //Mapper class
    public static class E_EMapper extends MapReduceBase implements
    Mapper<LongWritable, /*Input key Type */
    Text, /*Input value Type*/
    Text, /*Output key Type*/
    IntWritable> /*Output value Type*/
    {
        //Map function
        public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable>
output, Reporter reporter) throws IOException
        {
            String line = value.toString();
            String lasttoken = null;
            StringTokenizer s = new StringTokenizer(line, "\t");
            String year = s.nextToken();

            while(s.hasMoreTokens()){
                lasttoken=s.nextToken();
            }

            int avgprice = Integer.parseInt(lasttoken);
            output.collect(new Text(year), new IntWritable(avgprice));
        }
    }

    //Reducer class

    public static class E_EReduce extends MapReduceBase implements
    Reducer< Text, IntWritable, Text, IntWritable >
    {
        //Reduce function
        public void reduce(Text key, Iterator <IntWritable> values, OutputCollector<Text,
IntWritable> output, Reporter reporter) throws IOException
        {
            int maxavg=30;
            int val=Integer.MIN_VALUE;
            while (values.hasNext())
            {
                if((val=values.next().get())>maxavg)
                {
                    output.collect(key, new IntWritable(val));
                }
            }
        }
    }

    //Main function

    public static void main(String args[])throws Exception
    {
        JobConf conf = new JobConf(Eleunits.class);

        conf.setJobName("max_eletricityunits");

        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);

        conf.setMapperClass(E_EMapper.class);
        conf.setCombinerClass(E_EReduce.class);
        conf.setReducerClass(E_EReduce.class);

        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);

        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
    }
}

```

```
    JobClient.runJob(conf);  
  }  
}
```

Save the above program into **ProcessUnits.java**. The compilation and execution of the program is given below.

Compilation and Execution of ProcessUnits Program

Let us assume we are in the home directory of Hadoop user *e. g. /home/hadoop*.

Follow the steps given below to compile and execute the above program.

Step 1 – Use the following command to create a directory to store the compiled java classes.

```
$ mkdir units
```

Step 2 – Download Hadoop-core-1.2.1.jar, which is used to compile and execute the MapReduce program. Download the jar from mvnrepository.com. Let us assume the download folder is /home/hadoop/.

Step 3 – The following commands are used to compile the **ProcessUnits.java** program and to create a jar for the program.

```
$ javac -classpath hadoop-core-1.2.1.jar -d units ProcessUnits.java  
$ jar -cvf units.jar -C units/ .
```

Step 4 – The following command is used to create an input directory in HDFS.

```
$HADOOP_HOME/bin/hadoop fs -mkdir input_dir
```

Step 5 – The following command is used to copy the input file named **sample.txt** in the input directory of HDFS.

```
$HADOOP_HOME/bin/hadoop fs -put /home/hadoop/sample.txt input_dir
```

Step 6 – The following command is used to verify the files in the input directory

```
$HADOOP_HOME/bin/hadoop fs -ls input_dir/
```

Step 7 – The following command is used to run the Eleunit_max application by taking input files from the input directory.

```
$HADOOP_HOME/bin/hadoop jar units.jar hadoop.ProcessUnits input_dir output_dir
```

Wait for a while till the file gets executed. After execution, the output contains a number of input splits, Map tasks, Reducer tasks, etc.

```
INFO mapreduce.Job: Job job_1414748220717_0002  
completed successfully  
14/10/31 06:02:52  
INFO mapreduce.Job: Counters: 49
```

File System Counters

```
FILE: Number of bytes read=61  
FILE: Number of bytes written=279400  
FILE: Number of read operations=0  
FILE: Number of large read operations=0  
FILE: Number of write operations=0
```

```
HDFS: Number of bytes read=546  
HDFS: Number of bytes written=40
```

HDFS: Number of read operations=9
HDFS: Number of large read operations=0
HDFS: Number of write operations=2 Job Counters

Launched map tasks=2
Launched reduce tasks=1
Data-local map tasks=2

Total time spent by all maps in occupied slots (ms)=146137
Total time spent by all reduces in occupied slots (ms)=441
Total time spent by all map tasks (ms)=14613
Total time spent by all reduce tasks (ms)=44120

Total vcore-seconds taken by all map tasks=146137
Total vcore-seconds taken by all reduce tasks=44120

Total megabyte-seconds taken by all map tasks=149644288
Total megabyte-seconds taken by all reduce tasks=45178880

Map-Reduce Framework

Map input records=5
Map output records=5
Map output bytes=45
Map output materialized bytes=67

Input split bytes=208
Combine input records=5
Combine output records=5

Reduce input groups=5
Reduce shuffle bytes=6
Reduce input records=5
Reduce output records=5

Spilled Records=10
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2

GC time elapsed (ms)=948
CPU time spent (ms)=5160

Physical memory (bytes) snapshot=47749120
Virtual memory (bytes) snapshot=2899349504

Total committed heap usage (bytes)=277684224

File Output Format Counters

Bytes Written=40

Step 8 – The following command is used to verify the resultant files in the output folder.

```
$HADOOP_HOME/bin/hadoop fs -ls output_dir/
```

Step 9 – The following command is used to see the output in **Part-00000** file. This file is generated by HDFS.

```
$HADOOP_HOME/bin/hadoop fs -cat output_dir/part-00000
```

Following is the output generated by the MapReduce program –

1984 40

1985 45

Step 10 – The following command is used to copy the output folder from HDFS to the local file system.

```
$HADOOP_HOME/bin/hadoop fs -cat output_dir/part-00000/bin/hadoop dfs -get output_dir  
/home/hadoop
```

```
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul

Universitas
Esa Unggul



Modul 14

Distributed Web Based System



WEB SERVER

https://www.tutorialspoint.com/internet_technologies/web_servers.htm

Copyright © tutorialspoint.com

Overview

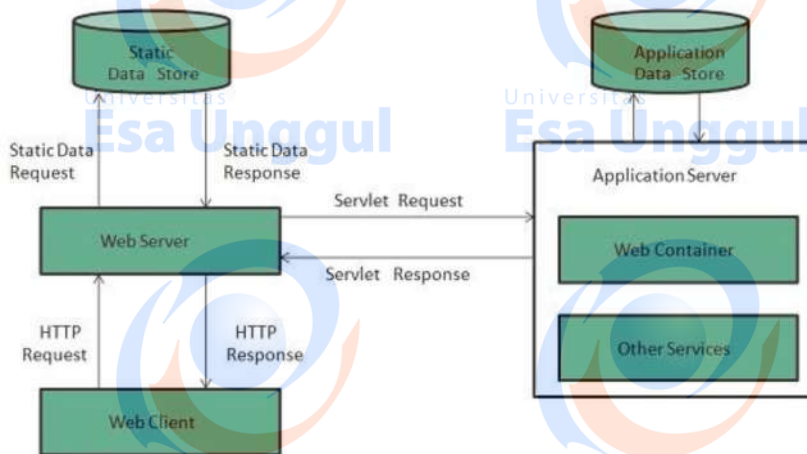
Web server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

| *Web site is collection of web pages while web server is a software that respond to the request for web resources.*

Web Server Working

Web server respond to the client request in either of the following two ways:

- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database



Key Points

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an **HTTP response:Error 404 Not found**.
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Architecture

Web Server Architecture follows the following two approaches:

1. Concurrent Approach
2. Single-Process-Event-Driven Approach.

Concurrent Approach

Concurrent approach allows the web server to handle multiple client requests at the same time. It can be achieved by following methods:

- Multi-process
- Multi-threaded
- Hybrid method.

Multi-processing

In this a single process *parent process* initiates several single-threaded child processes and distribute incoming requests to these child processes. Each of the child processes are responsible for handling single request.

It is the responsibility of parent process to monitor the load and decide if processes should be killed or forked.

Multi-threaded

Unlike Multi-process, it creates multiple single-threaded process.

Hybrid

It is combination of above two approaches. In this approach multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in single process results in less load on system resources.

Examples

Following table describes the most leading web servers available today:

S.N.	Web Server Description
1.	Apache HTTP Server This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.
2.	Internet Information Services IIS The Internet Information Server <i>IIS</i> is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms <i>and maybe on upcoming new Windows version also</i> . IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.
3.	Lighttpd The lighttpd, pronounced lighty is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.
4.	Sun Java System Web Server This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on Windows, Linux and UNIX platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, and Ruby on Rails, ASP and Coldfusion etc.
5.	Jigsaw Server Jigsaw <i>W3C's Server</i> comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, UNIX, Windows, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

INTERNET DOMAIN NAME SYSTEM

https://www.tutorialspoint.com/internet_technologies/internet_domain_name_system.htm

Copyright © tutorialspoint.com

Overview

When **DNS** was not into existence, one had to download a **Host file** containing host names and their corresponding IP address. But with increase in number of hosts of internet, the size of host file also increased. This resulted in increased traffic on downloading this file. To solve this problem the DNS system was introduced.

Domain Name System helps to resolve the host name to an address. It uses a hierarchical naming scheme and distributed database of IP addresses and associated names

IP Address

IP address is a unique logical address assigned to a machine over the network. An IP address exhibits the following properties:

- IP address is the unique address assigned to each host present on Internet.
- IP address is 32 bits *4bytes* long.
- IP address consists of two components: **network component** and **host component**.
- Each of the 4 bytes is represented by a number from 0 to 255, separated with dots. For example 137.170.4.124

| *IP address is 32-bit number while on the other hand domain names are easy to remember names. For example, when we enter an email address we always enter a symbolic string such as webmaster@tutorialspoint.com.*

Uniform Resource Locator URL

Uniform Resource Locator URL refers to a web address which uniquely identifies a document over the internet.

| *This document can be a web page, image, audio, video or anything else present on the web.*

For example, www.tutorialspoint.com/internet_technology/index.html is an URL to the index.html which is stored on tutorialspoint web server under internet_technology directory.

URL Types

There are two forms of URL as listed below:

1. Absolute URL
2. Relative URL

Absolute URL

Absolute URL is a complete address of a resource on the web. This completed address comprises of protocol used, server name, path name and file name.

For example http://www.tutorialspoint.com/internet_technology/index.htm. where:

- **http** is the protocol.
- **tutorialspoint.com** is the server name.
- **index.htm** is the file name.

The protocol part tells the web browser how to handle the file. Similarly we have some other protocols also that can be used to create URL are:

- FTP
- https
- Gopher
- mailto
- news

Relative URL

Relative URL is a partial address of a webpage. Unlike absolute URL, the protocol and server part are omitted from relative URL.

Relative URLs are used for internal links i.e. to create links to file that are part of same website as the WebPages on which you are placing the link.

For example, to link an image on `tutorialspoint.com/internet_technology/internet_reference_models`, we can use the relative URL which can take the form like `/internet_technologies/internet-osi_model.jpg`.

Difference between Absolute and Relative URL

Absolute URL	Relative URL
Used to link web pages on different websites	Used to link web pages within the same website.
Difficult to manage.	Easy to Manage
Changes when the server name or directory name changes	Remains same even if we change the server name or directory name.
Take time to access	Comparatively faster to access.

Domain Name System Architecture

The Domain name system comprises of **Domain Names, Domain Name Space, Name Server** that have been described below:

Domain Names

Domain Name is a symbolic string associated with an IP address. There are several domain names available; some of them are generic such as **com, edu, gov, net** etc, while some country level domain names such as **au, in, za, us** etc.

The following table shows the **Generic Top-Level Domain** names:

Domain Name	Meaning
Com	Commercial business
Edu	Education
Gov	U.S. government agency
Int	International entity
Mil	U.S. military
Net	Networking organization
Org	Non profit organization

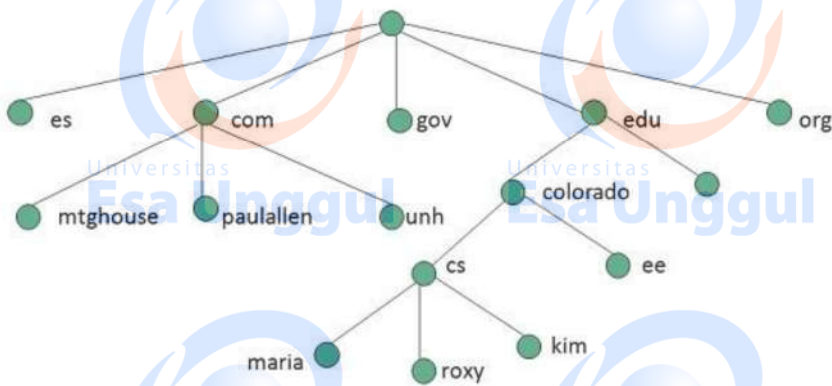
The following table shows the **Country top-level domain** names:

Domain Name	Meaning
au	Australia
in	India
cl	Chile
fr	France
us	United States
za	South Africa
uk	United Kingdom

jp	Japan
es	Spain
de	Germany
ca	Canada
ee	Estonia
hk	Hong Kong

Domain Name Space

The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels *from 0 to 127*, with a root at the top. The following diagram shows the domain name space hierarchy:



In the above diagram each subtree represents a domain. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

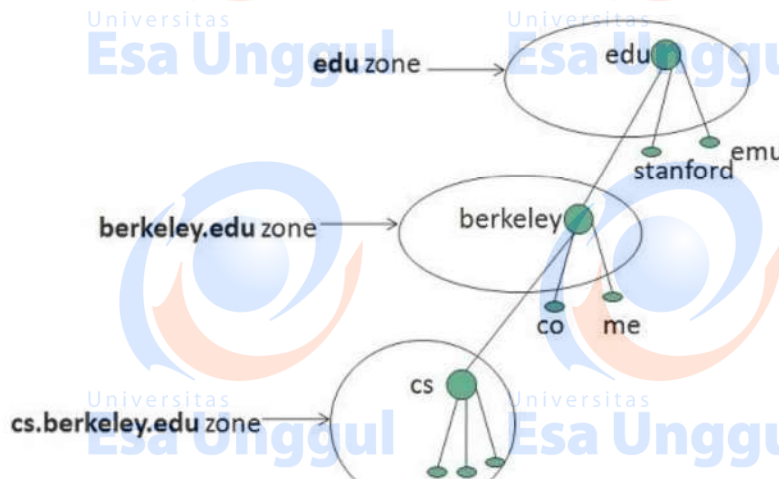
Name Server

Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.

- Hierarchy of server is same as hierarchy of names.
- The entire name space is divided into the zones

Zones

Zone is collection of nodes *subdomains* under the main domain. The server maintains a database called zone file for every zone.



| If the domain is not further divided into sub domains then domain and zone refers to the same thing.

The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers.

Types of Name Servers

Following are the three categories of Name Servers that manages the entire Domain Name System:

1. Root Server
2. Primary Server
3. Secondary Server

Root Server

Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server

Primary Servers

Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.

Secondary Server

Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.

DNS Working

DNS translates the domain name into IP address automatically. Following steps will take you through the steps included in domain resolution process:

- When we type **www.tutorialspoint.com** into the browser, it asks the local DNS Server for its IP address.
Here the local DNS is at ISP end.
- When the local DNS does not find the IP address of requested domain name, it forwards the request to the root DNS server and again enquires about IP address of it.
- The root DNS server replies with delegation that **I do not know the IP address of www.tutorialspoint.com but know the IP address of DNS Server.**
- The local DNS server then asks the com DNS Server the same question.
- The **com** DNS Server replies the same that it does not know the IP address of www.tutorialspoint.com but knows the address of tutorialspoint.com.
- Then the local DNS asks the tutorialspoint.com DNS server the same question.
- Then tutorialspoint.com DNS server replies with IP address of www.tutorialspoint.com.
- Now, the local DNS sends the IP address of www.tutorialspoint.com to the computer that sends the request.

INTERNET CONNECTIVITY

http://www.tutorialspoint.com/internet_technologies/internet_connectivity.htm

Copyright © tutorialspoint.com

Here in this tutorial, we will discuss how to connect to internet i.e. internet service providers, software and hardware requirements, configuring internet connection etc.

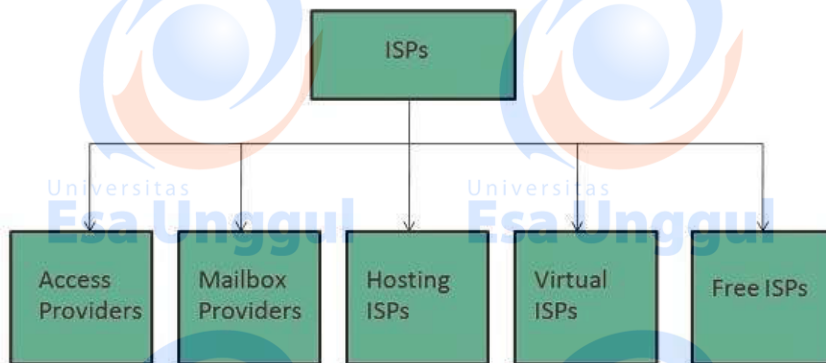
Internet Service Providers ISP

Internet Service Provider ISP is a company offering access to internet. They offer various services:

- Internet Access
- Domain name registration
- Dial-up access
- Leased line access

ISP Types

ISPs can broadly be classified into six categories as shown in the following diagram:



Access providers

They provide access to internet through telephone lines, cable wi-fi or fiber optics.

Mailbox Provider

Such providers offer mailbox hosting services.

Hosting ISPs

Hosting ISPs offers e-mail, and other web hosting services such as virtual machines, clouds etc.

Virtual ISPs

Such ISPs offer internet access via other ISP services.

Free ISPs

Free ISPs do not charge for internet services.

Connection Types

There exist several ways to connect to the internet. Following are these connection types available:

1. Dial-up Connection
2. ISDN
3. DSL
4. Cable TV Internet connections
5. Satellite Internet connections
6. Wireless Internet Connections

Dial-up Connection

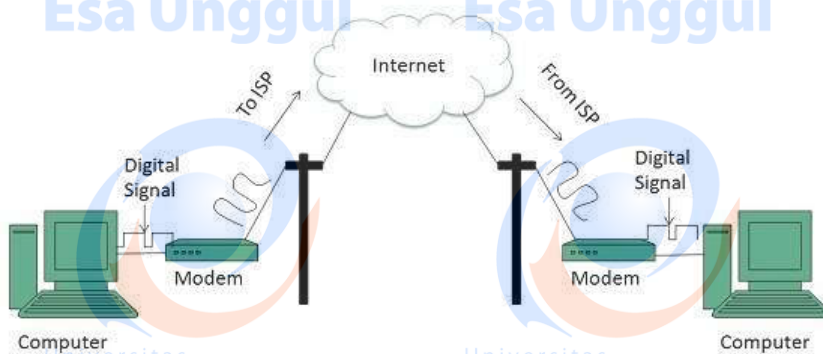
Dial-up connection uses telephone line to connect PC to the internet. It requires a modem to setup dial-up connection. This modem works as an interface between PC and the telephone line.

There is also a communication program that instructs the modem to make a call to specific number provided by an ISP.

Dial-up connection uses either of the following protocols:

1. Serial Line Internet Protocol *SLIP*
2. Point to Point Protocol *PPP*

The following diagram shows the accessing internet using modem:



ISDN

ISDN is acronym of **Integrated Services Digital Network**. It establishes the connection using the phone lines which carry digital signals instead of analog signals.

There are two techniques to deliver ISDN services:

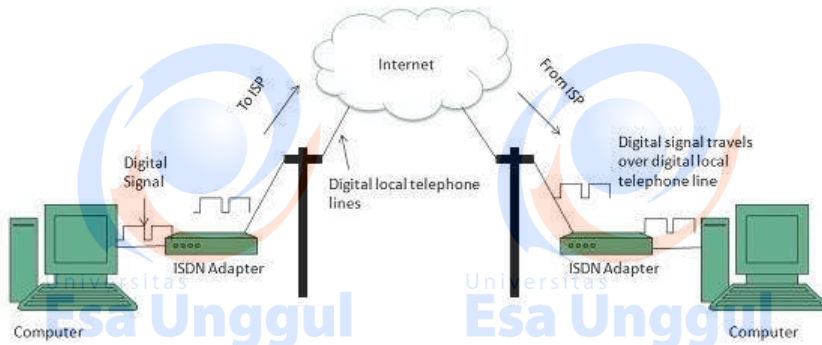
1. Basic Rate Interface *BRI*
2. Primary Rate Interface *PRI*

Key points:

- The BRI ISDN consists of three distinct channels on a single ISDN line: two 64kbps B *Bearer* channel and one 16kbps D *Delta or Data* channels.

- The PRI ISDN consists of 23 B channels and one D channels with both have operating capacity of 64kbps individually making a total transmission rate of 1.54Mbps.

The following diagram shows accessing internet using ISDN connection:



DSL

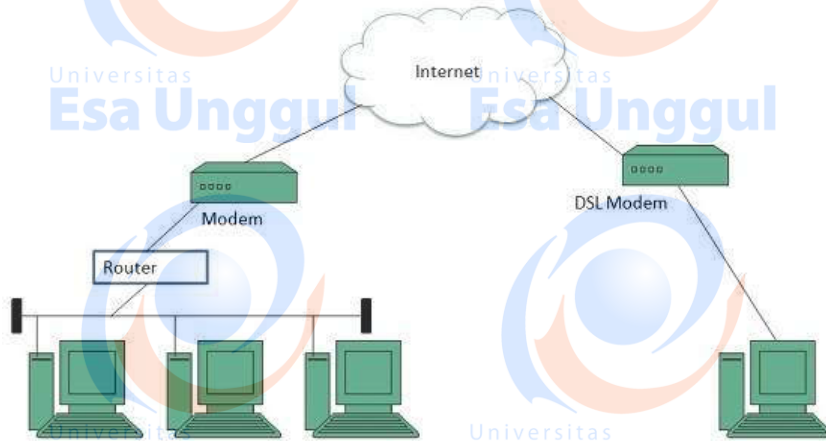
DSL is acronym of **Digital Subscriber Line**. It is a form of broadband connection as it provides connection over ordinary telephone lines.

Following are the several versions of DSL technique available today:

1. Asymmetric DSL *ADSL*
2. Symmetric DSL *SDSL*
3. High bit-rate DSL *HDSL*
4. Rate adaptive DSL *RDSL*
5. Very high bit-rate DSL *VDSL*
6. ISDN DSL *IDSL*

All of the above mentioned technologies differ in their upload and download speed, bit transfer rate and level of service.

The following diagram shows that how we can connect to internet using DSL technology:



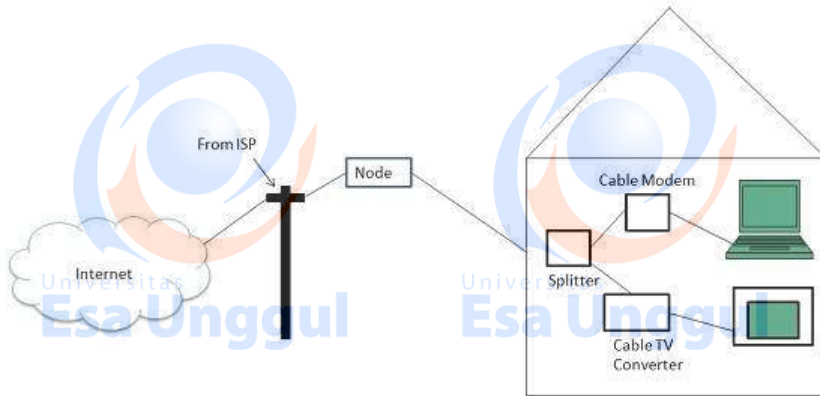
Cable TV Internet Connection

Cable TV Internet connection is provided through Cable TV lines. It uses coaxial cable which is capable of transferring data at much higher speed than common telephone line.

Key Points:

- A cable modem is used to access this service, provided by the cable operator.
- The Cable modem comprises of two connections: one for internet service and other for Cable TV signals.
- Since Cable TV internet connections share a set amount of bandwidth with a group of customers, therefore, data transfer rate also depends on number of customers using the internet at the same time.

The following diagram shows that how internet is accessed using Cable TV connection:



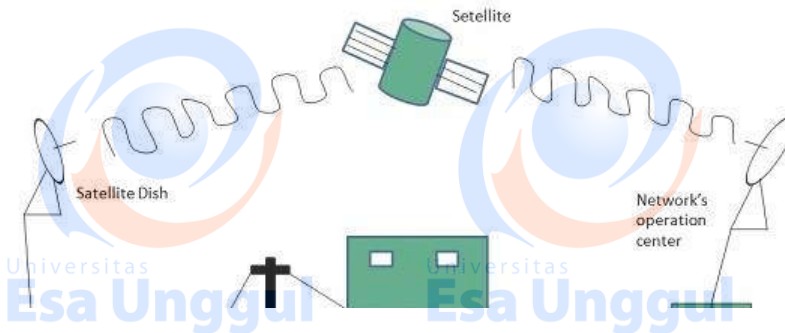
Satellite Internet Connection

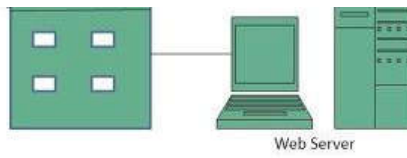
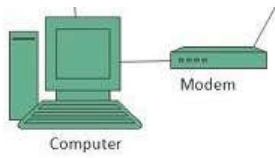
Satellite Internet connection offers high speed connection to the internet. There are two types of satellite internet connection: one way connection or two way connection.

In one way connection, we can only download data but if we want to upload, we need a dialup access through ISP over telephone line.

In two way connection, we can download and upload the data by the satellite. It does not require any dialup connection.

The following diagram shows how internet is accessed using satellite internet connection:





Wireless Internet Connection

Wireless Internet Connection makes use of radio frequency bands to connect to the internet and offers a very high speed. The wireless internet connection can be obtained by either WiFi or Bluetooth.

Key Points:

- Wi Fi wireless technology is based on IEEE 802.11 standards which allow the electronic device to connect to the internet.
- Bluetooth wireless technology makes use of short-wavelength radio waves and helps to create personal area network (PAN).

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js



WEBSITE SECURITY CONSIDERATION

https://www.tutorialspoint.com/internet_technologies/website_security.htm

Copyright © tutorialspoint.com

Websites are always to prone to security risks. **Cyber crime** impacts your business by hacking your website. Your website is then used for hacking assaults that install malicious software or malware on your visitor's computer.

Hackers may also steal important customer data such as credit card information, destroy your business and propagate illegal content to your users.

Security Considerations

Updated Software

It is mandatory to keep you software updated. It plays vital role in keeping your website secure.

SQL Injection

It is an attempt by the hackers to manipulate your database. It is easy to insert rogue code into your query that can be used to manipulate your database such as change tables, get information or delete data.

Cross Site Scripting XSS

It allows the attackers to inject client side script into web pages. Therefore, while creating a form It is good to endure that you check the data being submitted and encode or strip out any HTML.

Error Messages

You need to be careful about how much information to be given in the error messages. For example, if the user fails to log in the error message should not let the user know which field is incorrect: username or password.

Validation of Data

The validation should be performed on both server side and client side.

Passwords

It is good to enforce password requirements such as of minimum of eight characters, including upper case, lower case and special character. It will help to protect user's information in long run.

Upload files

The file uploaded by the user may contain a script that when executed on the server opens up your website.

SSL

It is good practice to use SSL protocol while passing personal information between website and web server or database.

SEARCH ENGINE OPTIMIZATION

http://www.tutorialspoint.com/internet_technologies/search_engine_optimization.htm Copyright © tutorialspoint.com

What is SEO

Search Engine Optimization refers to set of activities that are performed to increase number of desirable visitors who come to your site via search engine. These activities may include thing you do to your site itself, such as making changes to your text and HTML code, formatting text or document to communicate directly to the search engine.

Types of SEO

SEO can be classified into two types: **White Hat SEO** and **Black Hat or Spamdexing**

White Hat SEO

An SEO tactic, technique or method is considered as **White Hat** if it follows the followings

- If it conforms to the search engine's guidelines.
- If it does not involves any deception.
- It ensures that the content a search engine indexes and subsequently ranks is the same content a user will see.
- It ensures that a Web Page content should have been created for the users and not just for the search engines.
- It ensures the good quality of the web pages.
- It ensures the useful content available on the web pages.

Always follow a White Hat SEO tactic and don't try to fool your site visitors. Be honest and definitely you will get something more.

Black Hat or Spamdexing:

An SEO tactic, technique or method is considered as Black Hat or Spamdexing if it follows the followings:

- Try to improve rankings that are disapproved of by the search engines and/or involve deception.
- Redirecting users from a page that is built for search engines to one that is more human friendly.
- Redirecting users to a page that was different from the page the search engine ranked.
- Serving one version of a page to search engine spiders/bots and another version to human visitors. This is called **Cloaking SEO** tactic.
- Using Hidden or invisible text or with the page background color, using a tiny font size or hiding them within the HTML code such as "no frame" sections.
- Repeating keywords in the Meta tags, and using keywords that are unrelated to the site's content. This is called **Meta tag stuffing**.
- Calculated placement of keywords within a page to raise the keyword count, variety, and density of the page. This is called **Keyword stuffing**.

- Creating low-quality web pages that contain very little content but are instead stuffed with very similar key words and phrases. These pages are called **Doorway** or **Gateway Pages**
- **Mirror web sites** by hosting multiple web sites all with conceptually similar content but using different URLs.
- Mirror web sites by hosting multiple web sites all with conceptually similar content but using different URLs.
- Creating a rogue copy of a popular web site which shows contents similar to the original to a web crawler, but redirects web surfers to unrelated or malicious web sites. This is called **Page hijacking**.

Key Elements to ethical SEO

Keyword research

It allows you to see which keywords users actually employ to find products and services within your chosen market, instead of making guesses at the keywords you believe are the most popular.

Content development

Content development involves:

- Navigational flow and menu structure
- Site copy or articles
- Headings and sections

Web development

Web development involves:

- Text-based site development wherever possible.
- Clean and logical site structure.
- Proper markup of key page elements.

Link Building

Building links will make up about 60% of your work. There are ways to automate this process using shortcuts, workarounds, and submission services.

Internal linking is also very important. Treat the way you link to your own content same as you would link from an external site.

Webmaster Tools

Webmaster dashboard is provided by both **Google** and **Bing** that gives insight into activity by the search engine on any site that has been registered and verified via dashboard.

Dashboards offer a number of tools which allow us to understand how the search engine sees your site. These are the only way to **identify crawling, indexing, and the ranking issue** with your site.

SEO Do's and Do'nts

DO's

There are various other tips which can help you to optimize your web site for many Search Engines:

- Create logs of pages and each page should however contain a minimum of about 200 visible



words of text to maximize relevance with Google.

- Create a **Sitemap, Help, FAQ, About Us, Link to Us, Copyright, Disclaimer, and Privacy Policy** pages on mandatory basis.
- Create a home page link to each and every web page and provide easy navigation through all the pages.
- Pay attention to your dynamic page URLs. Google can crawl and index dynamic pages as long as you don't have more than 2 parameters in the URL.
- Check your complete site for broken links. Broken links will reduce your other pages rank as well.

Do'nts

Here is the list of Do'nts that one should keep in mind all the times:

- Don't keep hidden text on your web pages.
- Don't create alt image spamming by putting wrong keywords.
- Don't use Meta tags stuffing.
- Don't use frames and flash on your site.
- Don't exchange your links with black listed sites.
- Don't try to fool your site visitors by using miss spelled keywords.
- Don't send spam emails to thousand of email IDs.
- Don't use too much graphics on your site.
- Don't create too many doorway pages.
- Don't try to create duplicate content of pages.
- Don't submit your website many times in a single search engine.
- Don't use sub-directory depth more than 1-2.
- Don't create too many dynamic pages. Try to convert them into static pages.
- Don't bloat your pages with code.
- Don't nest your pages.



Modul 15

Distribution Coordinator Based System



WEB SERVER

https://www.tutorialspoint.com/internet_technologies/web_servers.htm

Copyright © tutorialspoint.com

Overview

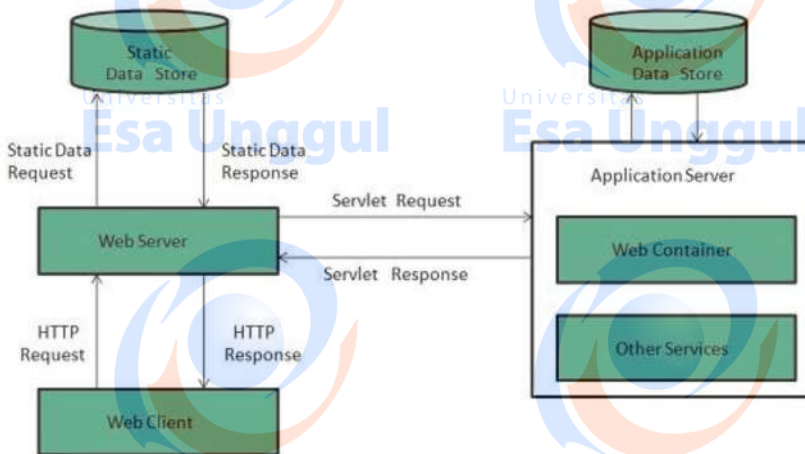
Web server is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

Web site is collection of web pages while web server is a software that respond to the request for web resources.

Web Server Working

Web server respond to the client request in either of the following two ways:

- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database



Key Points

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an **HTTP response:Error 404 Not found**.
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

Architecture

Web Server Architecture follows the following two approaches:

1. Concurrent Approach
2. Single-Process-Event-Driven Approach.

Concurrent Approach

Concurrent approach allows the web server to handle multiple client requests at the same time. It can be achieved by following methods:

- Multi-process
- Multi-threaded
- Hybrid method.

Multi-processing

In this a single process *parent process* initiates several single-threaded child processes and distribute incoming requests to these child processes. Each of the child processes are responsible for handling single request.

It is the responsibility of parent process to monitor the load and decide if processes should be killed or forked.

Multi-threaded

Unlike Multi-process, it creates multiple single-threaded process.

Hybrid

It is combination of above two approaches. In this approach multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in single process results in less load on system resources.

Examples

Following table describes the most leading web servers available today:

S.N.	Web Server Description
1.	Apache HTTP Server This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.
2.	Internet Information Services IIS The Internet Information Server <i>IIS</i> is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms and maybe on upcoming new Windows version also. IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.
3.	Lighttpd The lighttpd, pronounced lighty is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.
4.	Sun Java System Web Server This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on Windows, Linux and UNIX platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, and Ruby on Rails, ASP and Coldfusion etc.
5.	Jigsaw Server Jigsaw <i>W3C's Server</i> comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, UNIX, Windows, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

INTERNET DOMAIN NAME SYSTEM

https://www.tutorialspoint.com/internet_technologies/internet_domain_name_system.htm

Copyright © tutorialspoint.com

Overview

When **DNS** was not into existence, one had to download a **Host file** containing host names and their corresponding IP address. But with increase in number of hosts of internet, the size of host file also increased. This resulted in increased traffic on downloading this file. To solve this problem the DNS system was introduced.

Domain Name System helps to resolve the host name to an address. It uses a hierarchical naming scheme and distributed database of IP addresses and associated names

IP Address

IP address is a unique logical address assigned to a machine over the network. An IP address exhibits the following properties:

- IP address is the unique address assigned to each host present on Internet.
- IP address is 32 bits *4bytes* long.
- IP address consists of two components: **network component** and **host component**.
- Each of the 4 bytes is represented by a number from 0 to 255, separated with dots. For example 137.170.4.124

IP address is 32-bit number while on the other hand domain names are easy to remember names. For example, when we enter an email address we always enter a symbolic string such as webmaster@tutorialspoint.com.

Uniform Resource Locator URL

Uniform Resource Locator URL refers to a web address which uniquely identifies a document over the internet.

This document can be a web page, image, audio, video or anything else present on the web.

For example, www.tutorialspoint.com/internet_technology/index.html is an URL to the index.html which is stored on tutorialspoint web server under internet_technology directory.

URL Types

There are two forms of URL as listed below:

1. Absolute URL
2. Relative URL

Absolute URL

Absolute URL is a complete address of a resource on the web. This completed address comprises of protocol used, server name, path name and file name.

For example http://www.tutorialspoint.com/internet_technology/index.htm. where:

- **http** is the protocol.
- **tutorialspoint.com** is the server name.
- **index.htm** is the file name.

The protocol part tells the web browser how to handle the file. Similarly we have some other protocols also that can be used to create URL are:

- FTP
- https
- Gopher
- mailto
- news

Relative URL

Relative URL is a partial address of a webpage. Unlike absolute URL, the protocol and server part are omitted from relative URL.

Relative URLs are used for internal links i.e. to create links to file that are part of same website as the WebPages on which you are placing the link.

For example, to link an image on `tutorialspoint.com/internet_technology/internet_reference_models`, we can use the relative URL which can take the form like `/internet_technologies/internet-osi_model.jpg`.

Difference between Absolute and Relative URL

Absolute URL	Relative URL
Used to link web pages on different websites	Used to link web pages within the same website.
Difficult to manage.	Easy to Manage
Changes when the server name or directory name changes	Remains same even if we change the server name or directory name.
Take time to access	Comparatively faster to access.

Domain Name System Architecture

The Domain name system comprises of **Domain Names, Domain Name Space, Name Server** that have been described below:

Domain Names

Domain Name is a symbolic string associated with an IP address. There are several domain names available; some of them are generic such as **com, edu, gov, net** etc, while some country level domain names such as **au, in, za, us** etc.

The following table shows the **Generic** Top-Level Domain names:

Domain Name	Meaning
Com	Commercial business
Edu	Education
Gov	U.S. government agency
Int	International entity
Mil	U.S. military
Net	Networking organization
Org	Non profit organization

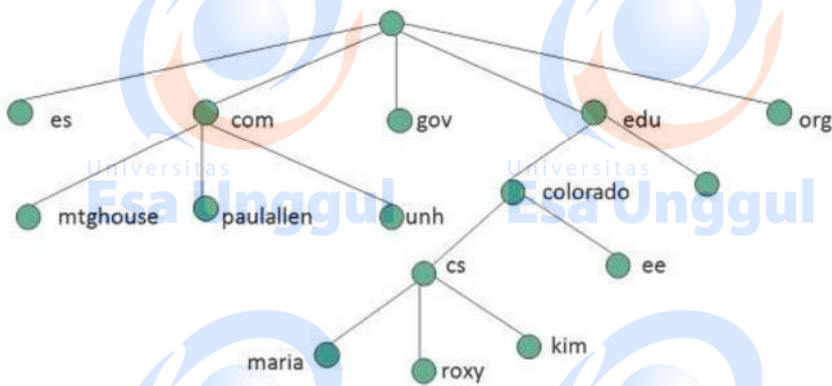
The following table shows the **Country top-level** domain names:

Domain Name	Meaning
au	Australia
in	India
cl	Chile
fr	France
us	United States
za	South Africa
uk	United Kingdom

jp	Japan
es	Spain
de	Germany
ca	Canada
ee	Estonia
hk	Hong Kong

Domain Name Space

The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels *from 0 to 127*, with a root at the top. The following diagram shows the domain name space hierarchy:



In the above diagram each subtree represents a domain. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

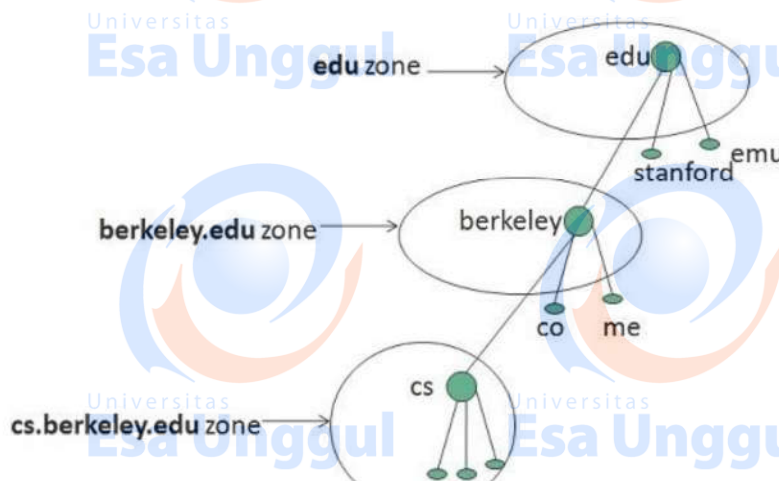
Name Server

Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.

- Hierarchy of server is same as hierarchy of names.
- The entire name space is divided into the zones

Zones

Zone is collection of nodes *subdomains* under the main domain. The server maintains a database called zone file for every zone.



| If the domain is not further divided into sub domains then domain and zone refers to the same thing.

The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers.

Types of Name Servers

Following are the three categories of Name Servers that manages the entire Domain Name System:

1. Root Server
2. Primary Server
3. Secondary Server

Root Server

Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server

Primary Servers

Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.

Secondary Server

Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.

DNS Working

DNS translates the domain name into IP address automatically. Following steps will take you through the steps included in domain resolution process:

- When we type **www.tutorialspoint.com** into the browser, it asks the local DNS Server for its IP address.
Here the local DNS is at ISP end.
- When the local DNS does not find the IP address of requested domain name, it forwards the request to the root DNS server and again enquires about IP address of it.
- The root DNS server replies with delegation that **I do not know the IP address of www.tutorialspoint.com but know the IP address of DNS Server.**
- The local DNS server then asks the com DNS Server the same question.
- The **com** DNS Server replies the same that it does not know the IP address of www.tutorialspoint.com but knows the address of tutorialspoint.com.
- Then the local DNS asks the tutorialspoint.com DNS server the same question.
- Then tutorialspoint.com DNS server replies with IP address of www.tutorialspoint.com.
- Now, the local DNS sends the IP address of www.tutorialspoint.com to the computer that sends the request.

INTERNET CONNECTIVITY

http://www.tutorialspoint.com/internet_technologies/internet_connectivity.htm

Copyright © tutorialspoint.com

Here in this tutorial, we will discuss how to connect to internet i.e. internet service providers, software and hardware requirements, configuring internet connection etc.

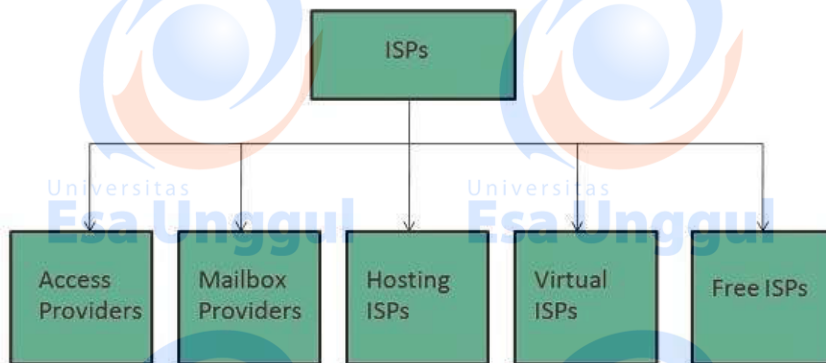
Internet Service Providers ISP

Internet Service Provider ISP is a company offering access to internet. They offer various services:

- Internet Access
- Domain name registration
- Dial-up access
- Leased line access

ISP Types

ISPs can broadly be classified into six categories as shown in the following diagram:



Access providers

They provide access to internet through telephone lines, cable wi-fi or fiber optics.

Mailbox Provider

Such providers offer mailbox hosting services.

Hosting ISPs

Hosting ISPs offers e-mail, and other web hosting services such as virtual machines, clouds etc.

Virtual ISPs

Such ISPs offer internet access via other ISP services.

Free ISPs

Free ISPs do not charge for internet services.

Connection Types

There exist several ways to connect to the internet. Following are these connection types available:

1. Dial-up Connection
2. ISDN
3. DSL
4. Cable TV Internet connections
5. Satellite Internet connections
6. Wireless Internet Connections

Dial-up Connection

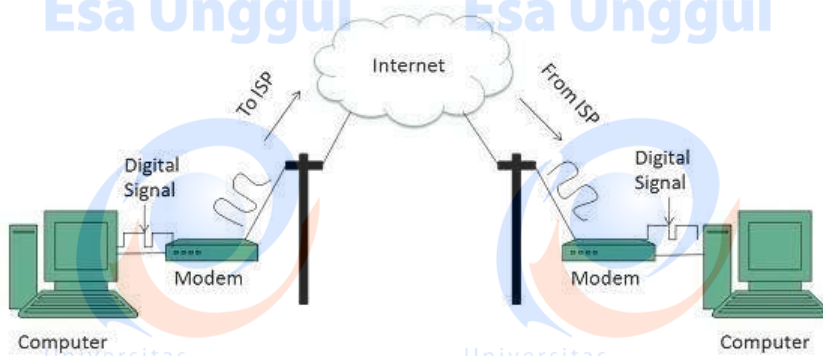
Dial-up connection uses telephone line to connect PC to the internet. It requires a modem to setup dial-up connection. This modem works as an interface between PC and the telephone line.

There is also a communication program that instructs the modem to make a call to specific number provided by an ISP.

Dial-up connection uses either of the following protocols:

1. Serial Line Internet Protocol *SLIP*
2. Point to Point Protocol *PPP*

The following diagram shows the accessing internet using modem:



ISDN

ISDN is acronym of **Integrated Services Digital Network**. It establishes the connection using the phone lines which carry digital signals instead of analog signals.

There are two techniques to deliver ISDN services:

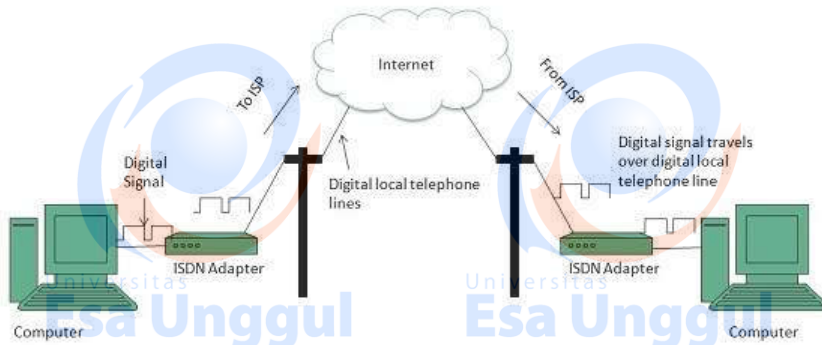
1. Basic Rate Interface *BRI*
2. Primary Rate Interface *PRI*

Key points:

- The BRI ISDN consists of three distinct channels on a single ISDN line: two 64kbps B *Bearer* channel and one 16kbps D *Delta or Data* channels.

- The PRI ISDN consists of 23 B channels and one D channels with both have operating capacity of 64kbps individually making a total transmission rate of 1.54Mbps.

The following diagram shows accessing internet using ISDN connection:



DSL

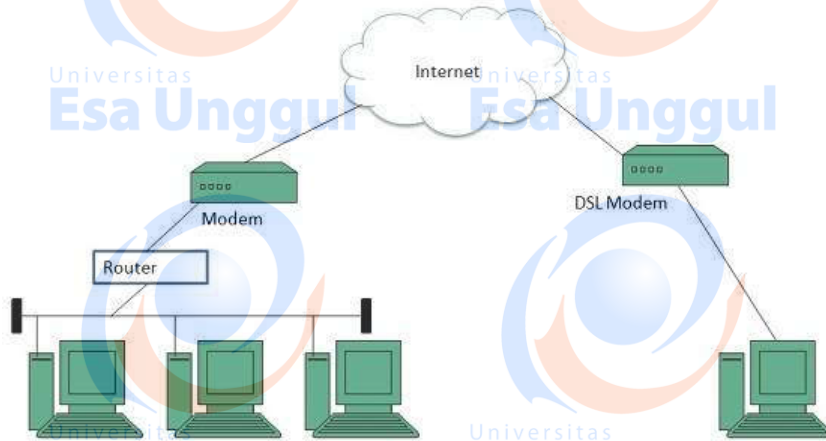
DSL is acronym of **Digital Subscriber Line**. It is a form of broadband connection as it provides connection over ordinary telephone lines.

Following are the several versions of DSL technique available today:

1. Asymmetric DSL *ADSL*
2. Symmetric DSL *SDSL*
3. High bit-rate DSL *HDSL*
4. Rate adaptive DSL *RDSL*
5. Very high bit-rate DSL *VDSL*
6. ISDN DSL *IDSL*

All of the above mentioned technologies differ in their upload and download speed, bit transfer rate and level of service.

The following diagram shows that how we can connect to internet using DSL technology:



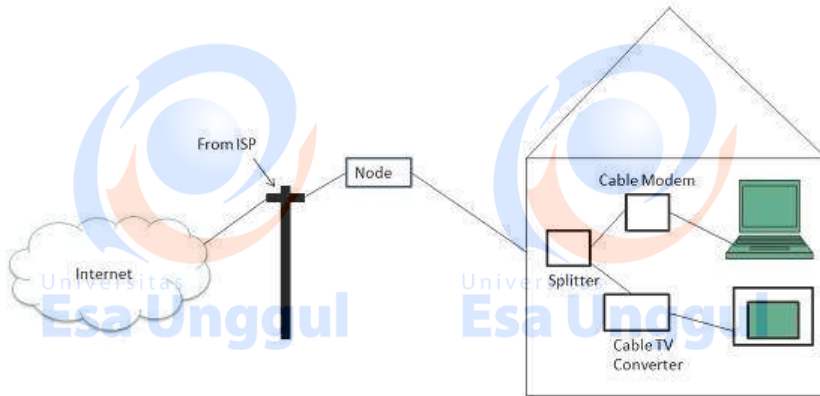
Cable TV Internet Connection

Cable TV Internet connection is provided through Cable TV lines. It uses coaxial cable which is capable of transferring data at much higher speed than common telephone line.

Key Points:

- A cable modem is used to access this service, provided by the cable operator.
- The Cable modem comprises of two connections: one for internet service and other for Cable TV signals.
- Since Cable TV internet connections share a set amount of bandwidth with a group of customers, therefore, data transfer rate also depends on number of customers using the internet at the same time.

The following diagram shows that how internet is accessed using Cable TV connection:



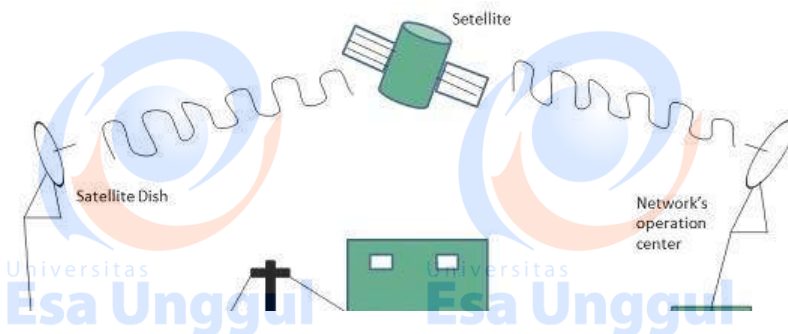
Satellite Internet Connection

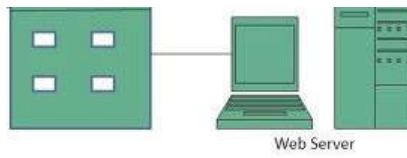
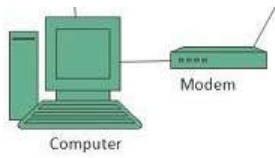
Satellite Internet connection offers high speed connection to the internet. There are two types of satellite internet connection: one way connection or two way connection.

In one way connection, we can only download data but if we want to upload, we need a dialup access through ISP over telephone line.

In two way connection, we can download and upload the data by the satellite. It does not require any dialup connection.

The following diagram shows how internet is accessed using satellite internet connection:





Wireless Internet Connection

Wireless Internet Connection makes use of radio frequency bands to connect to the internet and offers a very high speed. The wireless internet connection can be obtained by either WiFi or Bluetooth.

Key Points:

- Wi Fi wireless technology is based on IEEE 802.11 standards which allow the electronic device to connect to the internet.
- Bluetooth wireless technology makes use of short-wavelength radio waves and helps to create personal area network (PAN).

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js



WEBSITE SECURITY CONSIDERATION

https://www.tutorialspoint.com/internet_technologies/website_security.htm

Copyright © tutorialspoint.com

Websites are always to prone to security risks. **Cyber crime** impacts your business by hacking your website. Your website is then used for hacking assaults that install malicious software or malware on your visitor's computer.

Hackers may also steal important customer data such as credit card information, destroy your business and propagate illegal content to your users.

Security Considerations

Updated Software

It is mandatory to keep you software updated. It plays vital role in keeping your website secure.

SQL Injection

It is an attempt by the hackers to manipulate your database. It is easy to insert rogue code into your query that can be used to manipulate your database such as change tables, get information or delete data.

Cross Site Scripting XSS

It allows the attackers to inject client side script into web pages. Therefore, while creating a form It is good to endure that you check the data being submitted and encode or strip out any HTML.

Error Messages

You need to be careful about how much information to be given in the error messages. For example, if the user fails to log in the error message should not let the user know which field is incorrect: username or password.

Validation of Data

The validation should be performed on both server side and client side.

Passwords

It is good to enforce password requirements such as of minimum of eight characters, including upper case, lower case and special character. It will help to protect user's information in long run.

Upload files

The file uploaded by the user may contain a script that when executed on the server opens up your website.

SSL

It is good practice to use SSL protocol while passing personal information between website and web server or database.

SEARCH ENGINE OPTIMIZATION

http://www.tutorialspoint.com/internet_technologies/search_engine_optimization.htm Copyright © tutorialspoint.com

What is SEO

Search Engine Optimization refers to set of activities that are performed to increase number of desirable visitors who come to your site via search engine. These activities may include thing you do to your site itself, such as making changes to your text and HTML code, formatting text or document to communicate directly to the search engine.

Types of SEO

SEO can be classified into two types: **White Hat SEO** and **Black Hat or Spamdexing**

White Hat SEO

An SEO tactic, technique or method is considered as **White Hat** if it follows the followings

- If it conforms to the search engine's guidelines.
- If it does not involves any deception.
- It ensures that the content a search engine indexes and subsequently ranks is the same content a user will see.
- It ensures that a Web Page content should have been created for the users and not just for the search engines.
- It ensures the good quality of the web pages.
- It ensures the useful content available on the web pages.

Always follow a White Hat SEO tactic and don't try to fool your site visitors. Be honest and definitely you will get something more.

Black Hat or Spamdexing:

An SEO tactic, technique or method is considered as Black Hat or Spamdexing if it follows the followings:

- Try to improve rankings that are disapproved of by the search engines and/or involve deception.
- Redirecting users from a page that is built for search engines to one that is more human friendly.
- Redirecting users to a page that was different from the page the search engine ranked.
- Serving one version of a page to search engine spiders/bots and another version to human visitors. This is called **Cloaking SEO** tactic.
- Using Hidden or invisible text or with the page background color, using a tiny font size or hiding them within the HTML code such as "no frame" sections.
- Repeating keywords in the Meta tags, and using keywords that are unrelated to the site's content. This is called **Meta tag stuffing**.
- Calculated placement of keywords within a page to raise the keyword count, variety, and density of the page. This is called **Keyword stuffing**.

- Creating low-quality web pages that contain very little content but are instead stuffed with very similar key words and phrases. These pages are called **Doorway** or **Gateway Pages**
- **Mirror web sites** by hosting multiple web sites all with conceptually similar content but using different URLs.
- Mirror web sites by hosting multiple web sites all with conceptually similar content but using different URLs.
- Creating a rogue copy of a popular web site which shows contents similar to the original to a web crawler, but redirects web surfers to unrelated or malicious web sites. This is called **Page hijacking**.

Key Elements to ethical SEO

Keyword research

It allows you to see which keywords users actually employ to find products and services within your chosen market, instead of making guesses at the keywords you believe are the most popular.

Content development

Content development involves:

- Navigational flow and menu structure
- Site copy or articles
- Headings and sections

Web development

Web development involves:

- Text-based site development wherever possible.
- Clean and logical site structure.
- Proper markup of key page elements.

Link Building

Building links will make up about 60% of your work. There are ways to automate this process using shortcuts, workarounds, and submission services.

Internal linking is also very important. Treat the way you link to your own content same as you would link from an external site.

Webmaster Tools

Webmaster dashboard is provided by both **Google** and **Bing** that gives insight into activity by the search engine on any site that has been registered and verified via dashboard.

Dashboards offer a number of tools which allow us to understand how the search engine sees your site. These are the only way to **identify crawling, indexing, and the ranking issue** with your site.

SEO Do's and Do'nts

DO's

There are various other tips which can help you to optimize your web site for many Search Engines:

- Create logs of pages and each page should however contain a minimum of about 200 visible



words of text to maximize relevance with Google.

- Create a **Sitemap, Help, FAQ, About Us, Link to Us, Copyright, Disclaimer, and Privacy Policy** pages on mandatory basis.
- Create a home page link to each and every web page and provide easy navigation through all the pages.
- Pay attention to your dynamic page URLs. Google can crawl and index dynamic pages as long as you don't have more than 2 parameters in the URL.
- Check your complete site for broken links. Broken links will reduce your other pages rank as well.

Do'nts

Here is the list of Do'nts that one should keep in mind all the times:

- Don't keep hidden text on your web pages.
- Don't create alt image spamming by putting wrong keywords.
- Don't use Meta tags stuffing.
- Don't use frames and flash on your site.
- Don't exchange your links with black listed sites.
- Don't try to fool your site visitors by using miss spelled keywords.
- Don't send spam emails to thousand of email IDs.
- Don't use too much graphics on your site.
- Don't create too many doorway pages.
- Don't try to create duplicate content of pages.
- Don't submit your website many times in a single search engine.
- Don't use sub-directory depth more than 1-2.
- Don't create too many dynamic pages. Try to convert them into static pages.
- Don't bloat your pages with code.
- Don't nest your pages.