WWW (World Wide Web)

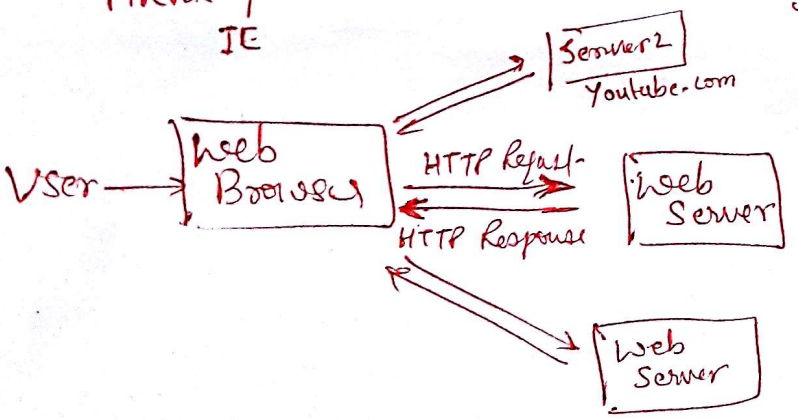
- Popularly known as web.
- Web began in 1989 at CERN
- Initially proposed by Tim Berners-Lee.
- Mosaic: was first graphical browser.
- 1990 – 2000 was known as dot com era. Among the millions of sites developed few popular ones are: Amazon, ebay, Google, facebook.
- 1994 – W3C (World Wide Web consortium) was created as an Organization to further develop the web, standardizing protocols and encouraging interoperability between sites.
- Facebook was started by Mark Zuckerberg.
- Google was started by Sergey Brin and Larry Page.

Architectural Overview:

→ a web — Consists of — Linked web Pages. — Linked via hyperlink (hypertext) (Invented by Vannevar Bush) — text / media

→ Browsers are used to view Pages. Browsers fetches page request, interprets the contents & display the properly formatted pages on the screen.

Firefox | IE | Chrome



Each page is fetched by sending a request to one or more servers.

→ Request Response protocol is used to fetch a page. This is simple text based protocol called HTTP which runs on TCP.

Page

- Static Page: Its contents remain same every time it is fetched.
- Dynamic Page: Its contents are generated on demand by a program.

CLIENT-SIDE

To identify a web Page :-

- Each Page is assigned a URL (Uniform Resource Locator)
- URL has 3 Parts
  - Protocol or scheme
  - DNS name of the machine on which page is located.
  - Path uniquely indicating specific page. (Hierarchical file Directory structure).

⇒ Steps carried out by browser to fetch a page when users clicks on a hyperlink :-

1. Browser determines URL
2. Browser asks DNS resolver for IP address of server.
3. DNS resolver replies IP address (eg- 128.208.00.88)
4. Browser makes TCP connection with 128.208.00.88 on port 80. (Port 80 is well known port for HTTP protocol).
5. It sends over an HTTP request asking for the page.
6. Server returns the page as HTTP response.
7. If page contents are to be fetched from other servers also then same process is repeated to Contact other Servers specified in URL.
8. Browser displays final page.
9. TCP Connections are released.

⇒ Few Common URL Schemes (or Protocols) :-

- http — hypertext (HTML)
- https - hypertext with security
- ftp - file transfer Protocol.
- file - Local file are accessed as a web page (No Server required)
- mailto - Sending e-mail.
- rtsp - establishing streaming media sessions.
- sip - audio/video multimedia calls.
- about - Provides information about the browser.
  eg about : plugins ⇒ will display list of MIME types

These schemes shows that URL has been designed not only to allow users to navigate the web, but to run older protocols such as FTP & email as well as newer protocols for audio & video and to provide Convenient access to local files & browser information.

SERVER SIDE

Steps performed by server :-

1. Accept a TCP Connection from a client (a browser).
2. Get name & path of the page requested.
3. Get the file (from disk) / for dynamic file program execution return the contents.
4. Send the Contents of the file to the client.
5. Release the TCP Connection.

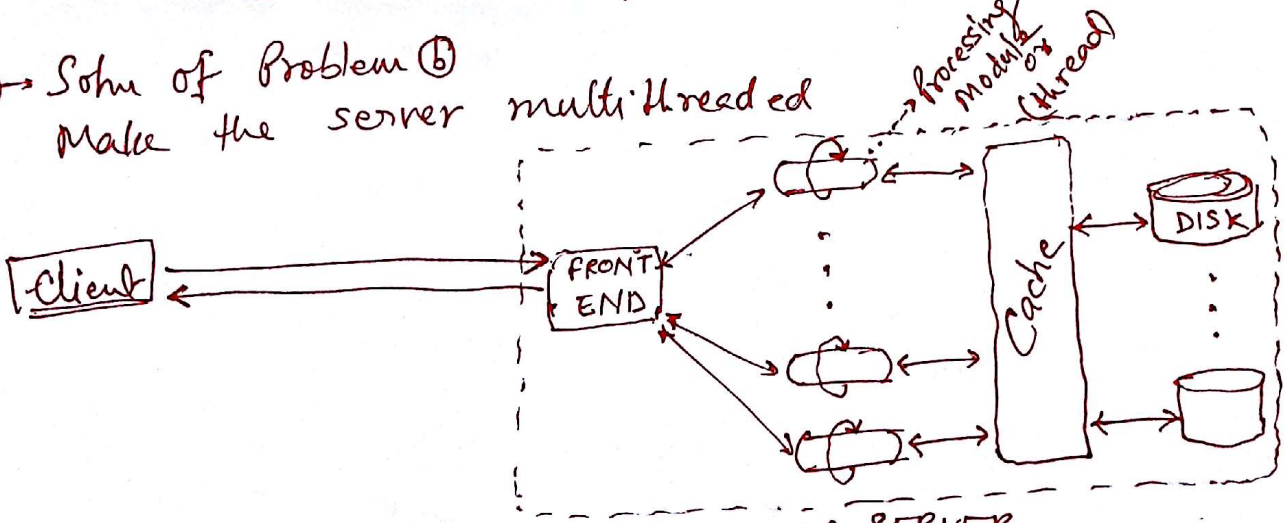
Limitation with Web Servers :-

(a) Disk reads are slow Compared to program execution and the same files may be read repeatedly from disk using OS Calls.
(b) Only one request is processed at a time. The file may be large & other request will be blocked while it is transferred.

Solution :-

(a) Solution of Problem (a) → Servers maintain a cache of n most recently read files (or a certain no. of gigabyte of Contents). ∴ Server can eliminate disk access if file is present on cache). This saves time although few overhead and expenses will be increased.

(b) → Soln of Problem (b)
Make the server multithreaded



MULTITHREADED WEB SERVER

(With front End and many processing modules).

- When a request comes in, the front end accepts it and builds a short record describing it, and then record is handed over to one of the processing modules. Processing module first checks cache for the availability of file, if it is not there it checks disk.
- Adv. of this scheme : while a processing module is blocked waiting for a disk or network operation to complete, other modules can work on other requests. With k processing modules, the throughput can be as much as k times higher than with a single threaded server.

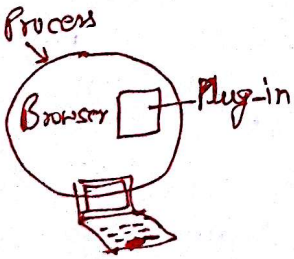
Despite all this, URL scheme has one inherent weakness :-
- A URL points to one specific host and sometimes it is required to refer a page without simultaneously telling where it is.
- Solution : URI (Uniform Resource Identifiers)

URI
- URL → tells how to locate a resource.
- URN (Uniform Resource Name) tells the name of a resource but not where to find it.

MIME :

To allow web browsers to understand format of web page to display any new page, web pages are written in a standardized language called HTML.
Apart from HTML, a web page may consist of MPEG video, PDF format document, JPEG format photograph, MP3 format song or hundreds of other types. In this case browser doesn't know how to interpret such pages.
When a server returns a page, it also returns some additional information ie, MIME type of the page. Browser maintains a table of MIME types to determine how to display the page.
There are 2 possibilities with browsers.

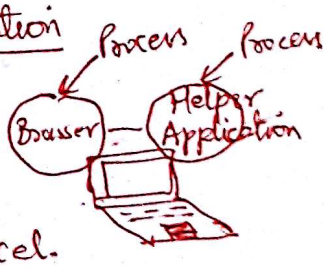
A Browser Plug-in



Third party module installed as an extension to the browser.
Plug-ins can access the current page and can modify its appearance ∵ Plug-ins runs inside the browser.
eg. Plug-ins for PDF, audio, video.

Helper Application

A Complete program running as a separate process, independently.
eg. MS-word, Excel.



→ Browsers can also open local files but need some way to determine the MIME type of the file. The standard method is for the OS to associate a file extension with a MIME type.

To reduce the complicated processing at each server, each processing module perform a series of steps:-

1. Resolve the name of web page requested.
2. Perform access control on the web page. ∵ all pages are not available to the general public.
3. Check the cache
4. Fetch the requested page from disk or run a program to build it.
5. Determine the rest of the response (eg. the MIME type).
6. Return the response to the client.
7. Make an entry in the server log for admin. purposes

COOKIES :→

→ When a server serves client request, it forgets that it has ever seen that particular client. In many cases it is required to keep record and behaviour of users (client). eg. newspaper, e-com, customized web portal.

→ Cookies can solve this problem.

→ Cookie is a small named string (at most 4 KB) that the server associate with the browser.

→ When a client requests a web page, the server supplies additional information in the form of cookies, along with the requested page.

→ Cookie are just strings not executable programs.

→ A cookie may contain upto 5 fields :-

Domain — Tells where the Cookie came from.

Path — / whole tree of the server's file tree

Content — name = value

Expires (Date & Time) This specifies when the Cookie expires.
- Persistent (If this field is absent, the browser discards the Cookie when it exits)
- Non-Persistent (If date & time are supplied) (GMT)

Secure (Y/N).

→ In many situations IP address can not be used to identify users. ∴ Cookies are required.

⇒ Cookies are commonly used to maintain the state of the session as a user browses around the site. The shopping cart is an example. You can place an item in the cart, switch to another page / site and when you come back the site knows who you are and you can continue with the order.
(This is required because HTTP is a stateless protocol ie, it doesn't remember the client).