Internet Technologies

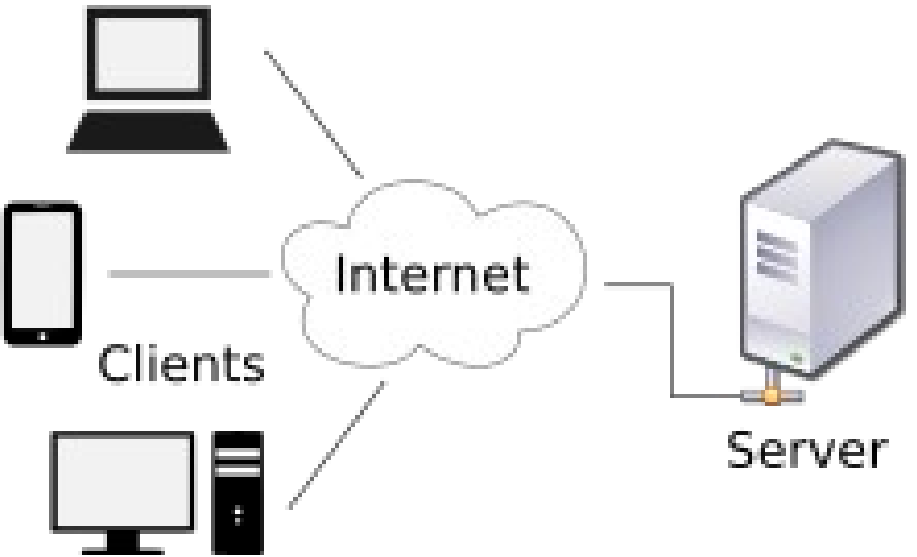
Web Servers

Mahesh Kumar
[maheshkumar@andc.du.ac.in]

Course Web Page
[www.mkbhandari.com/mkwiki]

Outline

1. Introduction to Web Servers
2. Working, Configuring, Hosting, and Managing Web Servers
3. Client-side Technologies
4. Server-side Technologies
5. Hybrid Technologies



PC clients connected to a web server via Internet **[1]**

Web Server

- The term web server can refer to hardware or software, or both of them working together.

1. *On the hardware side*

   → *a web server is a computer that stores web server software and a website's component files (for example, HTML documents, images, CSS stylesheets, and JavaScript files)*

   → *a web server connects to the Internet and supports physical data interchange with other devices connected to the web.*

Web Server

- The term web server can refer to hardware or software, or both of them working together.

2. *On the software side*

   → *a web server includes several parts that control how web users access hosted files. At a minimum, this is an HTTP server*

   → *an HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages).*

   → *an HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device.*

Web Server

- At the most basic level, whenever a browser needs a file that is hosted on a web server, the browser requests the file via HTTP
- When the request reaches the correct (hardware) web server, the (software) HTTP server accepts the request, finds the requested document, and sends it back to the browser, also through HTTP
- If the server doesn't find the requested document, it returns a 404 response instead.

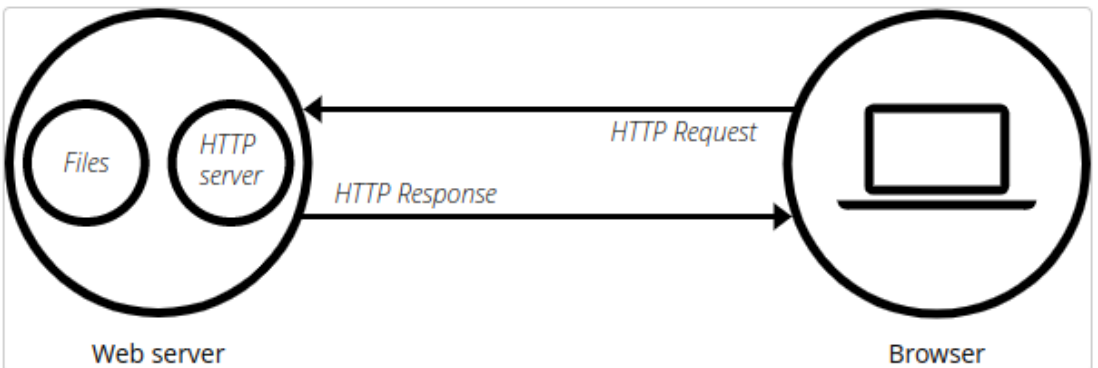


Figure 1: HTTP Request and HTTP Response **[2]**

HTTP and HTML: Berners-Lee's Basics

- HTTP is a communication standard governs the requests and responses that take place between the browser running on the end user's computer and the web server.
- The server's job is to accept a request from the client (*the web browser and the computer on which it's running)* and attempt to reply to it in a meaningful way, usually by serving up a requested web page
- Between the client and the server there can be several other devices, such as routers, proxies, gateways, and so on.
  - *Serves different roles in ensuring that the requests and responses are correctly transferred between the client and server.*
  - *Typically, they use the Internet to send this information.*

HTTP and HTML: Berners-Lee's Basics

- A web server can usually handle multiple simultaneous connections and — when not communicating with a client — spends its time listening for an incoming connection.
- When one arrives, the server sends back a response to confirm its receipt.

The Request/Response Procedure

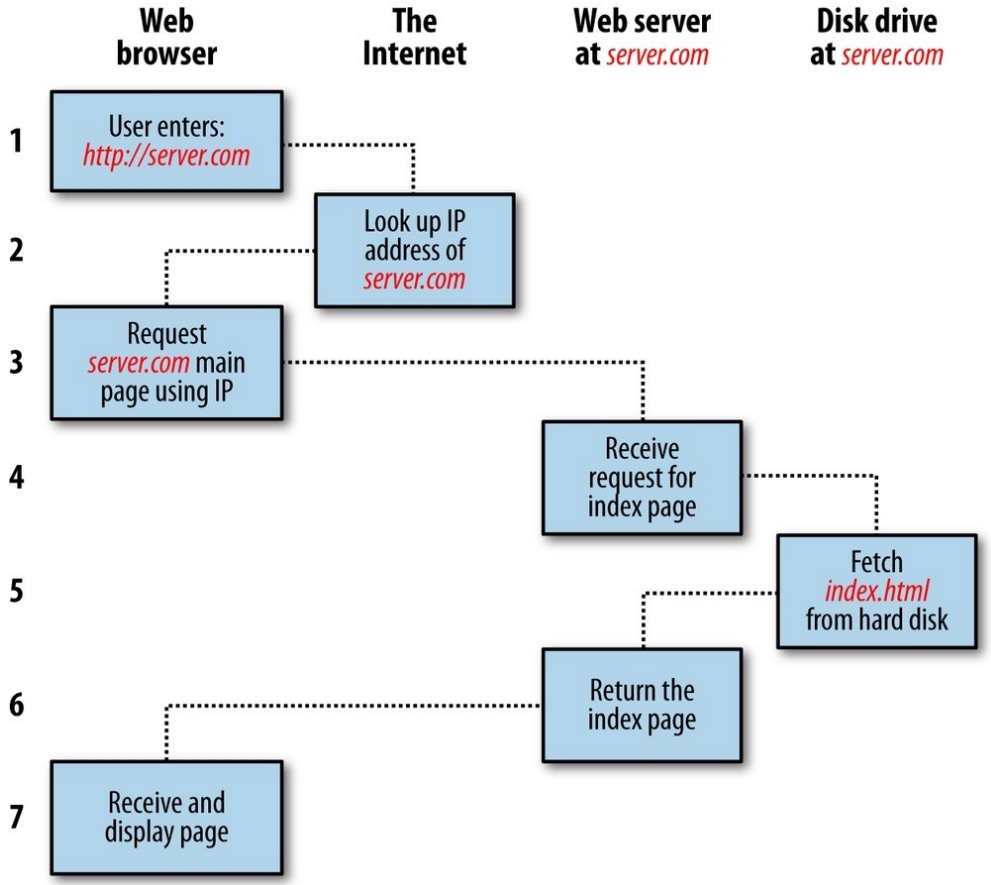


Figure 2: The basic client/server request/response sequence **[3]**

The Request/Response Procedure

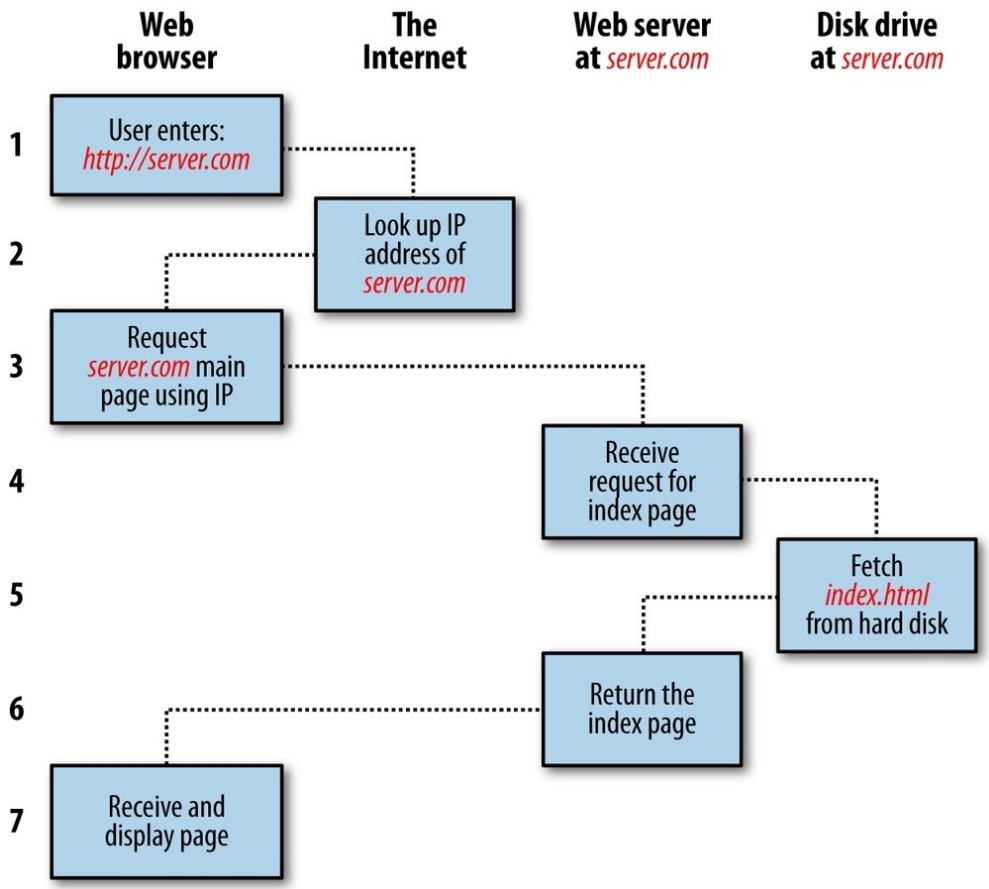


Figure 2: The basic client/server request/response sequence **[3]**

1. *You enter http://server.com into your browser's address bar.*
2. *Your browser looks up the IP address for server.com.*
3. *Your browser issues a request for the home page at server.com.*
4. *The request crosses the Internet and arrives at the server.com web server.*
5. *The web server, having received the request, looks for the web page on its hard disk.*

The Request/Response Procedure

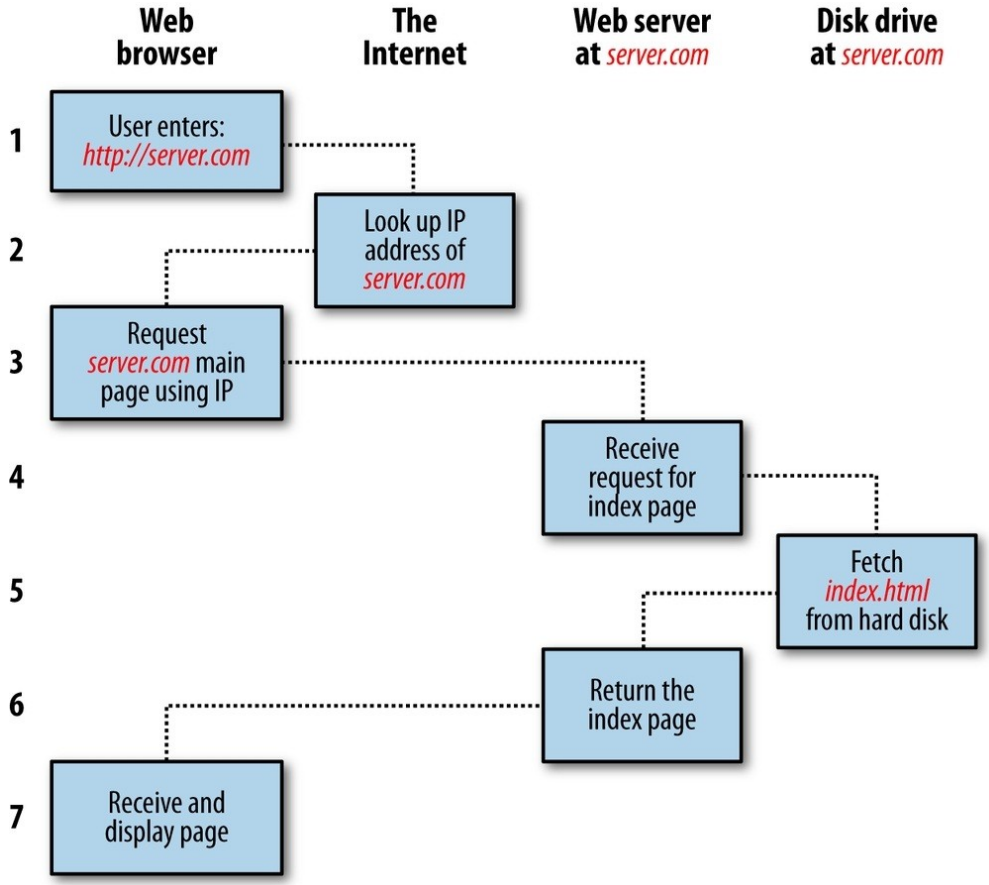


Figure 2: The basic client/server request/response sequence **[3]**

6. *The web page is retrieved by the server and returned to the browser.*

7. *Your browser displays the web page.*

The Request/Response Procedure

- For an average web page(static web page), this process takes place once for each object within the page: a graphic, an embedded video or Flash file, and even a CSS template.
- In step 2, the browser looked up the IP address of server.com.
  - *every machine attached to the Internet has an IP address*
  - *but we generally access web servers by name, such as google.com.*
  - *the browser consults an additional Internet service called the Domain Name Service (DNS) to find its associated IP address and then uses it to communicate with the computer.*
- For dynamic web pages, the procedure is a little more involved, because it may bring both PHP and MySQL into the mix (see Figure 3).

The Request/Response Procedure

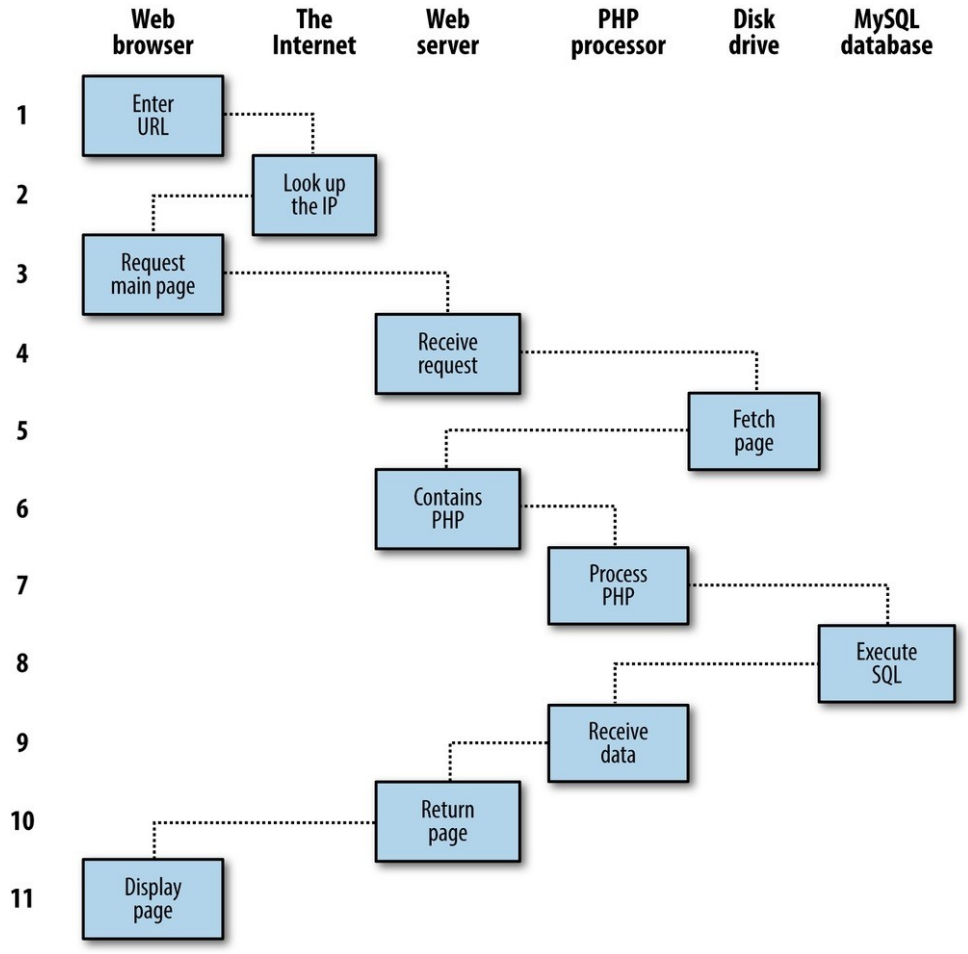


Figure 3: A dynamic client/server request/response sequence **[3]**

The Request/Response Procedure

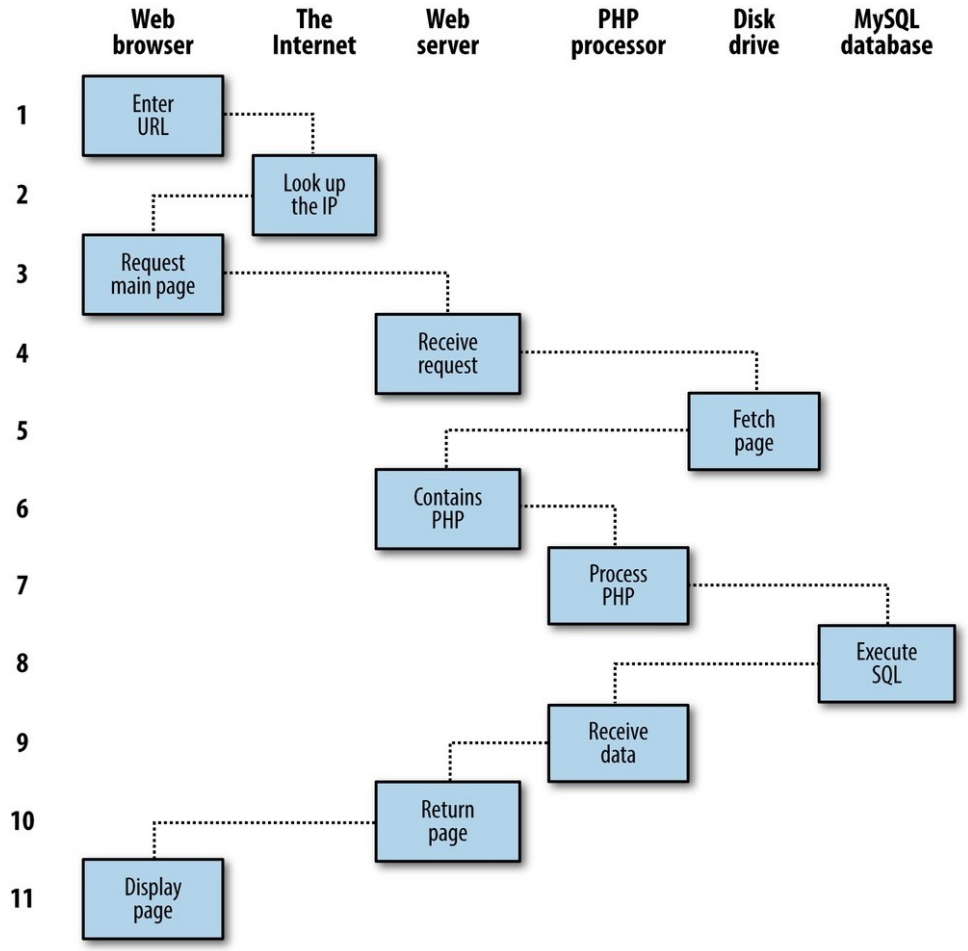


Figure 3: A dynamic client/server request/response sequence **[3]**

1. *You enter http://server.com into your browser's address bar.*
2. *Your browser looks up the IP address for server.com.*
3. *Your browser issues a request for the home page at server.com.*
4. *The request crosses the Internet and arrives at the server.com web server.*
5. *The web server, having received the request, looks for the web page on its hard disk.*

The Request/Response Procedure

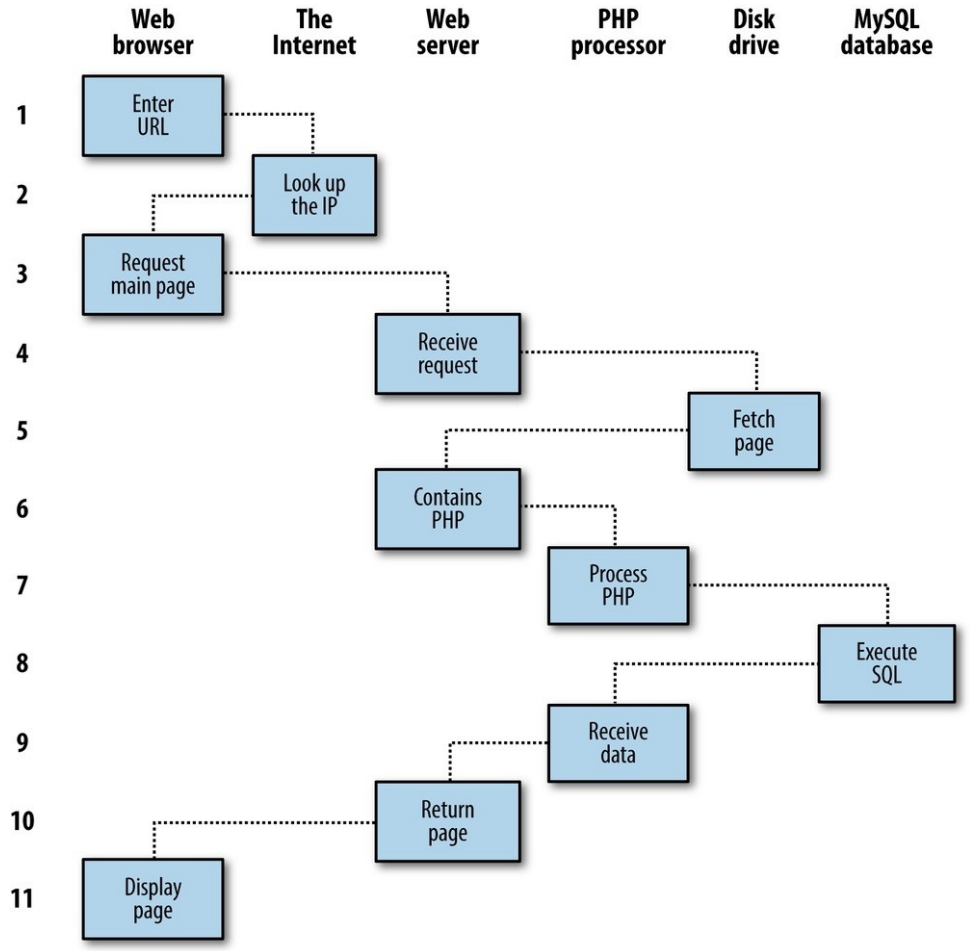


Figure 3: A dynamic client/server request/response sequence **[3]**

6. *With the home page now in memory, the web server notices that it is a file incorporating PHP scripting and passes the page to the PHP interpreter.*

7. *The PHP interpreter executes the PHP code.*

8. *Some of the PHP contains MySQL statements, which the PHP interpreter now passes to the MySQL database engine.*

9. *The MySQL database returns the results of the statements back to the PHP interpreter.*

The Request/Response Procedure

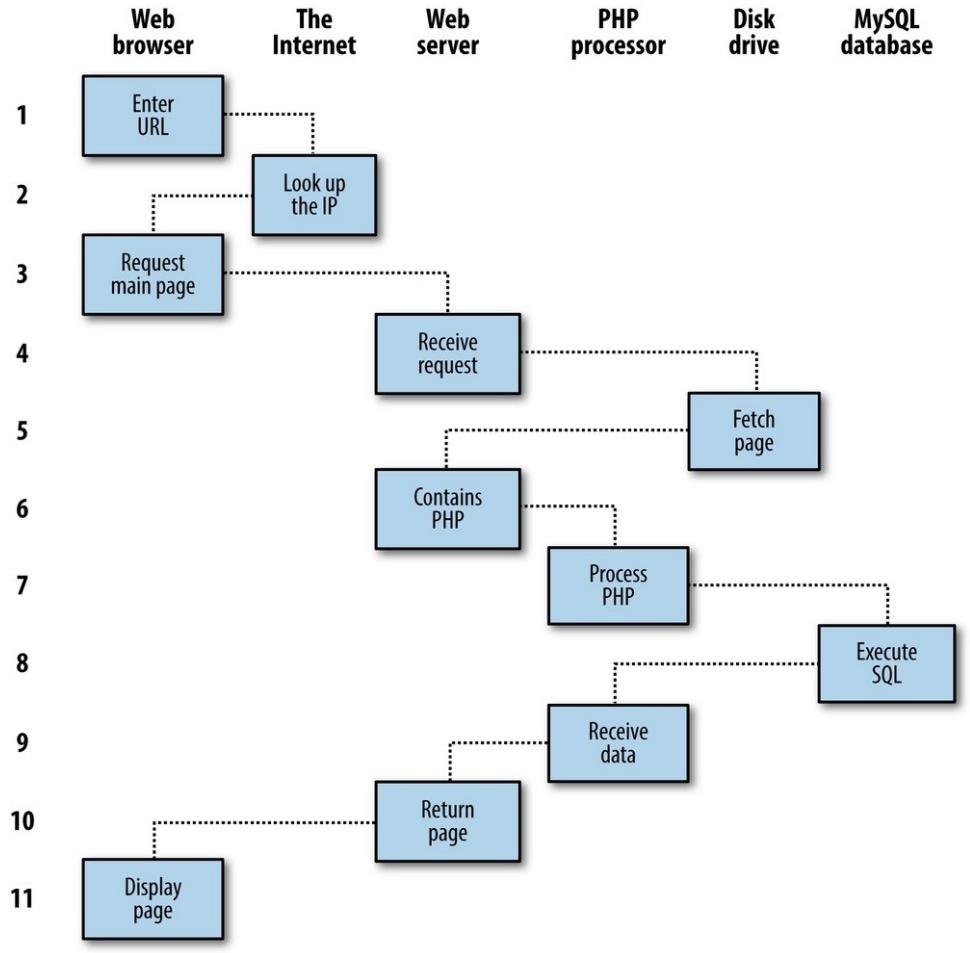


Figure 3: A dynamic client/server request/response sequence **[3]**

10 *The PHP interpreter returns the results of the executed PHP code, along with the results from the MySQL database, to the web server.*

11 *The web server returns the page to the requesting client, which displays it.*

PHP

- PHP is a widely-used, free, server side scripting language, and a powerful tool for making dynamic and interactive Web pages.
  - *With PHP, it's a simple matter to embed dynamic activity in web pages*
  - *When you give pages the .php extension, they have instant access to the scripting language.*
  - *From a developer's point of view, all you have to do is write code such as the following:*

```
<?php
      echo " Today is " . date("l") . ". ";
?>
Here's the latest news.
```

1. *The opening **<?php** tells the web server to allow the PHP program to interpret all the following code up to the **?>** tag.*
2. *Outside of this construct, everything is sent to the client as direct HTML.*

PHP

- *The final output of the two parts looks like this:*

  Today is Tuesday. Here's the latest news.

- *PHP is a flexible language, you can also place the PHP construct directly next to PHP code, like this:*

  Today is <?php echo date("l"); ?>. Here's the latest news.

- Using PHP, you have unlimited control over your web server.
  - *modify HTML on the fly*
  - *process a credit card*
  - *add user details to a database*
  - *fetch information from a third-party website*

  you can do it all from within the same PHP files in which the HTML itself resides.

MySQL

- MySQL is free, open-source and a widely used relational database management system (RDBMS).
  - *ideal for both small and large applications.*
  - *installed on vast numbers of Internet web servers is a robust and exceptionally fast database management system that uses English-like commands.*
- The highest level of MySQL structure is a database, within which you can have one or more tables that contain your data.
  - *let's suppose you are working on a table called users, within which you have created columns for surname, firstname, and email, and you now wish to add another user. One command that you might use to do this is:*

```
INSERT INTO users VALUES('Smith', 'John', 'jsmith@mysite.com');
```

MySQL

- *It's equally easy to look up data. Let's assume that you have an email address for a user and need to look up that person's name.*

  SELECT surname,firstname FROM users WHERE email='jsmith@mysite.com';

  MySQL will then return Smith, John and any other pairs of names that may be associated with that email address in the database.

- There's quite a bit more that you can do with MySQL than just simple INSERT and SELECT commands.
  - *join multiple tables according to various criteria*
  - *ask for results in a variety of orders*
  - *Make partial matches when you know only part of the string that you are searching for*
  - *return only the nth result, and a lot more.*

MySQL

- Using PHP, you can make all these calls directly to MySQL without having to run the MySQL program yourself or use its command-line interface.

- you can save the results in arrays for processing and perform multiple lookups, each dependent on the results returned from earlier ones, to drill right down to the item of data you need.

JavaScript

- JavaScript provides a means for dynamic user interaction, such as:
  - *checking email address validity in input forms*
  - *displaying prompts, and so on*
- Combined with CSS, JavaScript is the power behind dynamic web pages that change in front of your eyes rather than when a new page is returned by the server.
- let's take a quick look at how you can use basic JavaScript, accepted by all browsers:

```
<script type="text/javascript">
      document.write("Today is " + Date() );
</script>
```

This code snippet tells the web browser to interpret everything within the script tags as JavaScript, which the browser then does by writing the text Today is to the current document, along with the date, by using the JavaScript function Date.

JavaScript

- JavaScript was originally developed to offer dynamic control over the various elements within an HTML document, but more and more, JavaScript is being used for Ajax
  - *Ajax: accesses the web server in the background.*
  - *a quick Ajax call can pull in and update a single element on a web page, such as changing your photograph on a social networking site or replacing a button that you click with the answer to a question.*

CSS

- With CSS you can style any HTML element to change its dimensions, colors, borders, spacing, add animated transitions and transformations to your web pages.
- Using CSS can be as simple as inserting a few rules between <style> and </style> tags in the head of a web page, like this:

```
<style>
    p {
        text-align:justify;
        font-family:Helvetica;
    }
</style>
```

These rules will change the default text alignment of the <p> tag so that paragraphs contained in it will be fully justified and will use the Helvetica font.

- There are many different ways you can lay out CSS rules.

HTML5

- A great deal of new stuff in HTML (and quite a few things that have been changed or removed), but in summary, here's what you get:
  - *Markup*
    - → *Including new elements such as <nav> and <footer>, and deprecated elements like <font> and .*
  - *New APIs*
    - → *For example, the <canvas> element for writing and drawing on a graphics canvas, <audio> and <video> elements, offline web apps, microdata, and local storage*
  - *Applications*
    - → *Including two new rendering technologies: MathML (Math Markup Language) for displaying mathematical formulae and SVG (Scalable Vector Graphics) for creating graphical elements outside of the new <canvas> element.*

The Apache Web Server

- In addition to PHP, MySQL, JavaScript, CSS, and HTML5, there's actually a sixth hero in the dynamic Web: the web server.
- Apache doesn't serve up just HTML files — it handles a wide range of files from images and Flash files to MP3 audio files, and so on.
  - *each element a web client encounters in an HTML page is also requested from the server, which then serves it up.*
  - *But these objects don't have to be static files such as GIF images. They can all be generated by programs such as PHP scripts*
  - *PHP can even create images and other files for you, either on the fly or in advance to serve up later.*
    - → *modules either precompiled into Apache or PHP or called up at runtime. For example GD (Graphics Draw) library, which PHP uses to create and handle graphics.*

The Apache Web Server

- Apache also supports a huge range of modules of its own.
  - *In addition to the PHP module, the most important for your purposes as a web programmer are the modules that handle security.*
  - *the Rewrite module, which enables the web server to handle a varying range of URL types and rewrite them to its own internal requirements*
  - *the Proxy module, which you can use to serve up often-requested pages from a cache to ease the load on the server.*

About Open Source

- PHP, MySQL, and Apache are the three most commonly used tools in their categories.
- All three are open source, means that they have been developed in the community by teams of programmers writing the features they themselves want and need, with the original code available for all to see and change. Bugs can be found and security breaches can be prevented before they happen.
- Another benefit: all these programs are free to use. There's no worrying about having to purchase additional licenses if you have to scale up your website and add more servers. And you don't need to check the budget before deciding whether to upgrade to the latest versions of these products.

Bringing It All Together

- The real beauty of PHP, MySQL, JavaScript, CSS, and HTML5 is the wonderful way in which they all work together to produce dynamic web content:
  - *PHP handles all the main work on the web server*
  - *MySQL manages all the data*
  - *the combination of CSS and JavaScript looks after web page presentation.*
  - *JavaScript can also talk with your PHP code on the web server whenever it needs to update something (either on the server or on the web page)*
  - *The powerful new features in HTML5, such as the canvas, audio and video, and geolocation, you can make your web pages highly dynamic, interactive, and multimedia packed.*
  - *With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behaviour of the existing page.*

Configuring, Hosting, and Managing Web Servers

- Class Assignment

References

1. https://en.wikipedia.org/wiki/Web_server
2. https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server
3. Learning PHP, MySQL, JavaScript, CSS & HTML5, (Robin Nixon), 3rd Edition, O'Reilly Media