# Web Architecture

Dylan Lane McDonald

CNM STEMulus Center
Web Development with PHP

January 8, 2016

# Outline

1. Client/Server Model
   - Client
   - Server
2. xAMP Stack
   - What Is it?
   - Operating Systems
   - Apache
   - mySQL
   - PHP
3. Open Source
4. HTTP Conversation

# Client/Server Model

The client/server model is a classic model that describes multiple small nodes (clients) connecting to and utilizing a centralized node (server). The key idea of the client/server model is all the data is the same no matter what type of client is utilizing the server.

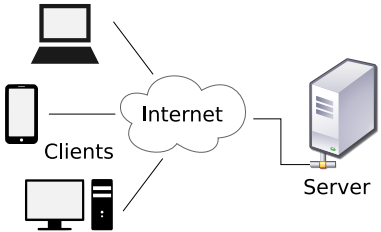


Figure 1: Client/Server Model

# Client

In terms of the web, the client is a **web browser**. A web browser consists of three major components:

- **HTML Renderer**: Responsible for interpreting HTML code and determining where elements are placed on a page and how they show up
- **CSS Layout Engine**: Responsible for interpreting CSS and applying specified styles to the matched HTML elements
- **JavaScript Engine**: Responsible for interpreting JavaScript and executing JavaScript in response to client side events

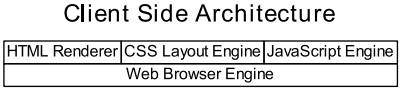


Figure 2: Client Side Architecture

# Server

The server is a machine configured with multiple **dæmons**[1], processes that run in an infinite loop replying to incoming requests from clients. Servers typically have very robust CPU, disk, and network configurations. This allows servers to do the "heavy lifting" on behalf on clients.

In this class, we will concentrate on the *x*AMP architecture.

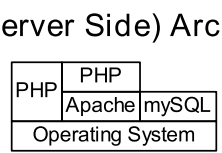


Figure 3: Server Side Architecture

# What Is The *x*AMP Stack?

As previously discussed, the *x*AMP stack is the combination of:

- **A**pache
- **m**ySQL
- **P**HP

# What Is The *x*AMP Stack?

As previously discussed, the *x*AMP stack is the combination of:

- **A**pache
- **m**ySQL
- **P**HP

The word "stack" is a computer science term for data layered one upon the other. The name implies that *x* (the operating system) is the lower layer, the next layer is Apache, the following layer is mySQL, and the last layer is PHP. This is mostly true. One cannot exist without the layer below it.

# Operating Systems

The lowest layer of the stack is the operating system. The most common operating systems used are:

- "UNIX"?
  - Linux[2]
  - BSD
  - Solaris
- Mac OS X[3]
- Windows

No matter what the operating system, as long as the other components are installable, one can create an *x*AMP stack.

[2] Linux is **technically** not UNIX. It's a UNIX clone.
[3] Actually, Mac OS X is more UNIX than Linux is.

# Apache Web Server

The Apache web server is the most popular web server today, with a 38% market share. [1] The name Apache is derived from the myriad of "patches" needed to improve its predecessor, NCSA `httpd` to fix bugs and extend features. Within a year, it became the internet's most popular web server. [2]

# Apache Web Server

The Apache web server is the most popular web server today, with a 38% market share. [1] The name Apache is derived from the myriad of "patches" needed to improve its predecessor, NCSA `httpd` to fix bugs and extend features. Within a year, it became the internet's most popular web server. [2]

The purpose of the Apache web server is to listen on the HTTP port (port 80) and receive and reply to HTTP commands such as, "give me the page `index.php`" or "Take this data and give it to the page `form_process.php`".

# mySQL

mySQL is a **R**elational **D**atabase **M**anagement **S**ystem (RDBMS) is is "officially pronounced 'My Ess Que Ell' (not 'my sequel')". [3] mySQL and PostgreSQL are the two most popular open source RDMBSes.

The reference version of mySQL is 5.6. The differences in the 5.$x$ versions of mySQL, while not trivial, are smaller when compared in the version differences in PHP.

# PHP

PHP (**P**HP **H**ypertext **P**rocessor) is a server side scripting language that has evolved into a larger general purpose programming language. The current version is 7. PHP remained at 5.*x* for 11 years and PHP 5.*x* code, especially PHP 5.5 and 5.6 code are still in active use.

PHP is a highly extensible language with multiple extensions installable server wide. In addition, individual projects can be extended by using `*.phar` files (a mnemonic for "PHP archive") without the necessity of deploying system wide changes to the PHP configuration. The Compser[4] project also allows one to add additional features to PHP without deploying system wide changes to the PHP configuration.

[4] Composer is akin to Java's Maven

# Open Source

All the components of the *x*AMP stack, as well as some of the possible operating systems are open source.

# Open Source

All the components of the *x*AMP stack, as well as some of the possible operating systems are open source.

## Quote of the Day

*"When I released GNU Emacs and people started using it, they started sending me improvements in the mail. So I would get a message with a bug fix, and a message with a new feature, and another bug fix, and another new feature, and another... and another... until they were pouring in on me so fast that just taking advantage of all of the help people were giving me was a big job. Microsoft doesn't have this problem."*
$\approx$ *Richard Stallman*

# Open Source

The open source movement was, among other people, started by Richard Stallman when he worked at the artificial intelligence lab at MIT in the 1970s.

# Open Source

The open source movement was, among other people, started by Richard Stallman when he worked at the artificial intelligence lab at MIT in the 1970s.

## Vitæ

**Richard Stallman** *is the founder of the GNU Project and Free Software Foundation. He also pioneered the "copyleft" movement to legally protect free software. He remains a visible (and somewhat controversial) figure in the software industry and dedicates his life to the free software movement.*

# HTTP Conversation

Informally, everything on the web happens in an **HTTP Conversation**. This conversation is typically initiated by the client and consists of one or more replies from the server. Extending the conversation metaphor, such a conversation consists of the following steps:

- **Client**: Hi, server! I need to get `foo.php`.
- **Server**: Hi, client! Here's `foo.php`. Can I get anything else for you?
- **Client**: Yes, `bar.php`.
- **Server**: Here's `bar.php`. Can I get anything else for you?
- **Client**: No thank you. Goodbye.

The conversation will continue until the client has requested all the items required.

# HTTP Request

## Anatomy of an HTTP Request

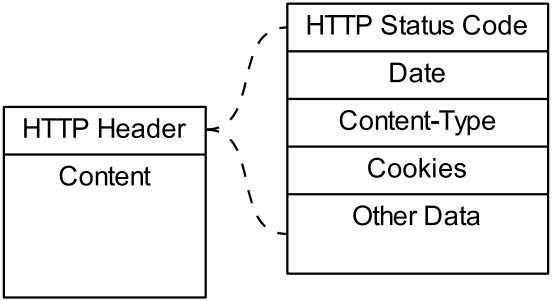


Figure 4: HTTP Request

# HTTP Request

Figure 4 depicts the anatomy of the conversation outlined in Slide 13. Notice there are two slices of the reply from the server: the header and the content. The **header** contains the metadata[5] about the request.

The header contains a lot about the request. Of particular interest in the header is:

- **HTTP Status Code**: a 3 digit code that indicates the result:
  - 1XX: incomplete, but not yet failed
  - 2XX: complete, successful
  - 3XX: complete, somewhere else
  - 4XX: error, blame the client
  - 5XX: error, blame the server
- **Content Type**: what type of data the request produced

[5] "data about data"

# Static HTTP Conversation

Returning to the conversation in Slide 13, what if the server doesn't need to conduct database queries, contact other servers, etc and is able to grab all content internally? This is a known as a **static use case**. A static use case would sound something like this:

- **Client**: Hi, server! I need to get `static.php`.
- **Server**: Hi, client! Here's `static.php`. Can I get anything else for you?
- **Client**: No thank you. Goodbye.

Notice no extra steps are required by the server in the static use case.

# Dynamic HTTP Conversation

Now suppose the server does need to conduct database queries, contact other servers, etc and some content relies on external data. This is a known as a **dynamic use case**. A dynamic use case would sound something like this:

- **Client**: Hi, server! I need to get `dynamic.php`.
- **Server**: Hi, client! One moment while I search my database and contact my partners...
- **Server**: ...thanks for waiting for me to gather external data. Here's `dynamic.php`. Can I get anything else for you?
- **Client**: No thank you. Goodbye.

Notice the server has to delay and wait for the external data to become available in the dynamic use case.

# Works Cited


Netcraft Ltd.
March 2015 web server survey.
http://news.netcraft.com/archives/2015/03/19/march-2015-web-server-survey.html.


Apache Foundation.
About apache.
http://httpd.apache.org/ABOUT_APACHE.html.


Oracle Corporation.
What is mysql?
http://dev.mysql.com/doc/refman/5.6/en/what-is-mysql.html.