Introduction to Web Development
Week 1 - Intro, The Internet!, & Dev Environment

Dr. Paul Talaga
487 Rhodes
paul.talaga@uc.edu
ACM Lecture Series
University of Cincinnati, OH

October 10, 2012
About the Series

Goals

- High-level understanding of web technologies
- What makes up a website.
- How you can build & maintain one.
- Whatever interests you!

Schedule

Week 1  Intro, How the Internet Works, Dev Environment
Week 2  HTML, CSS, PHP
Week 3  Graphics!
Week 4  Page Layout
Week 5  SEO, JavaScript, AJAX, ...
Week 6  CMS - Wordpress, Joomla!, Drupal, etc...
Old Class Website

http://fuzzpault.com/instruction/tfta.php

Me

- BS in Math & CS from St.Lawrence University. ’03
- MS in CS from Syracuse University. ’06
- Ph.D from Syracuse University 2012
  Exploiting Data Locality in Dynamic Web Applications
- University of Cincinnati - Assistant Professor Educator
- ”Modeling user interactions for (fun and) profit: preventing request forgery attacks on web applications” - 2009 - Jayaraman, Talaga, Lewandowski, Chapin, Hafiz
- ”Enforcing request integrity in web applications” - 2010 - Jayaraman, Lewandowski, Talaga, Chapin
- Design, Maintain, & Host 9 sites (starting 1996).
Sizes

**Bit** 0/1 (b)

**Byte** 8 bits (B) \(2^8 = 256\) \(\approx\) 1 character

- **Kilobyte** kB - 1000 bytes \(2^{10} = 1024\) bytes memory, KiB
- **Megabyte** MB - \(10^6\) bytes \(2^{20} = 1024\) kB memory, MiB
- **Gigabyte** GB - \(10^9\) bytes \(2^{30} = 1024\) MB memory, GiB
- **Terabyte** TB - \(10^{12}\) bytes \(2^{40} = 1024\) GB memory, TiB
- **Petabyte** PB - You get the idea.

Speeds: Latency - Bandwidth

- Latency (ms) \(10^{-3}\) seconds
- kb/s, kbit/s, kbps - kilobit per second (b not B!)
- Mb/s, Mbit/s, Mbps - Megabit per second
- Gb/s, Gbit/s, Gbps - Gigabit per second
### Sizes

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
<th>Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>0/1 (b)</td>
<td></td>
</tr>
<tr>
<td>Byte</td>
<td>8 bits (B) (2^8 = 256) ≈ 1 character</td>
<td></td>
</tr>
<tr>
<td>Kilobyte</td>
<td>kB - 1000 bytes (2^10 = 1024 bytes memory, KiB)</td>
<td></td>
</tr>
<tr>
<td>Megabyte</td>
<td>MB - 10^6 bytes (2^20 = 1024 kB memory, MiB)</td>
<td></td>
</tr>
<tr>
<td>Gigabyte</td>
<td>GB - 10^9 bytes (2^30 = 1024 MB memory, GiB)</td>
<td></td>
</tr>
<tr>
<td>Terabyte</td>
<td>TB - 10^12 bytes (2^40 = 1024 GB memory, TiB)</td>
<td></td>
</tr>
<tr>
<td>Petabyte</td>
<td>PB - You get the idea.</td>
<td></td>
</tr>
</tbody>
</table>

### Speeds: Latency - Bandwidth

- Latency (ms) (10^-3 seconds)
- kb/s, kbit/s, kbps - kilobit per second (b not B!)
- Mb/s, Mbit/s, Mbps - Megabit per second
- Gb/s, Gbit/s, Gbps - Gigabit per second
The Internet!

What is the Internet?

"It’s a series of tubes." - Ted Stevens, 2006 - NO!

Jargon

Web  Interconnected networks of computers
IP   Internet Protocol v4 & v6
TCP  Transmission Control Protocol
DNS  Domain Name System
HTTP(S)  Hypertext Transfer Protocol
HTML Hypertext Markup Language
URL  Uniform Resource Locator
What is the Internet?
"It’s a series of tubes." - Ted Stevens, 2006 - NO!

**Jargon**

- **Web**: Interconnected networks of computers
- **IP**: Internet Protocol v4 & v6
- **TCP**: Transmission Control Protocol
- **DNS**: Domain Name System
- **HTTP(S)**: Hypertext Transfer Protocol
- **HTML**: Hypertext Markup Language
- **URL**: Uniform Resource Locator
IPv4 vs IPv6

IPv4 32bit, IPv6 128bit
4 billion in IPv4, World Population 6 billion!
Domain Name System: Maps domain name to IP address

PING google.com (74.125.225.71): 56 data bytes
64 bytes from 74.125.225.71: time=16.638 ms

In NJ. Why not the same?!?
PING google.com (74.125.137.102) 56 bytes of data.
64 bytes from (74.125.137.102): time=23.5 ms

PING uc.edu (10.23.135.100): 56 data bytes
64 bytes from 10.23.135.100: time=1.130 ms

Every website must have a nameserver responding to DNS requests.
Putting it together

HowStuffWorks.com

How the Internet Works

©2010 HowStuffWorks

Web Server

Domain Name Server

Router

Packet

ISP

Request Path

Return Path

Modem
HTTP?

HyperText Transfer Protocol

- Request-response protocol
- http - TCP port 80
  - https http through SSL/TLS TCP port 443
- HTTP/1.0 - 3 methods: GET, POST, HEAD
- HTTP/1.1 - 8 methods total
- Status Codes: 200, 400, 404
- Single TCP session per request, HTTP/1.1 adds persistent connection
- A Webserver speaks HTTP!
- SPDY: An experimental protocol for a faster web
HTTP?

HyperText Transfer Protocol

- Request-response protocol
- http - TCP port 80
- https http through SSL/TLS TCP port 443
- HTTP/1.0 - 3 methods: GET, POST, HEAD
- HTTP/1.1 - 8 methods total
- Status Codes: 200, 400, 404
- Single TCP session per request, HTTP/1.1 adds persistent connection
- A Webserver speaks HTTP!
- SPDY: An experimental protocol for a faster web
Web Server Market Share

Netcraft.com

Graph showing the market share of web servers over time. The graph includes data from various companies such as Apache, Microsoft, Sun, nginx, Google, NCSA, and Other. The x-axis represents the years from 1995 to 2012, with specific months labeled. The y-axis represents the market share percentage.
Development Environment: This Works for Me

Setup:
- Fedora 14 - Similar to CentOS (Webserver)
- Local Webserver & Database (Apache w/PHP, MySQL)
- Editors (Emacs, gedit, Gimp, Photoshop)
- Web Browsers (Chrome, Firefox, IE, Opera, Safari)

Development Progression:
1. Write HTML/PHP
2. View from local server
3. Repeat until happy
4. View in other browsers/ benchmark
5. Deploy! (FTP, SFTP, SSH, SCP, rsync)
6. Weekly/Monthly snapshots of local files
7. Weekly Database snapshots from webserver
Development Environment: This Works for Me

Setup:
- Fedora 14 - Similar to CentOS (Webserver)
- Local Webserver & Database (Apache w/PHP, MySQL)
- Editors (Emacs, gedit, Gimp, Photoshop)
- Web Browsers (Chrome, Firefox, IE, Opera, Safari)

Development Progression:
1. Write HTML/PHP
2. View from local server
3. Repeat until happy
4. View in other browsers/ benchmark
5. Deploy! (FTP, SFTP, SSH, SCP, rsync)
6. Weekly/Monthly snapshots of local files
7. Weekly Database snapshots from webserver
Thanks! See you next week!