

# Web Hosting

Michael Tsai

2015/05/25

# Web Hosting Basics

- A daemon (server) listens for connection on TCP port 80
- Accept request for documents
- Transmits them to the requesting user's browser

# URL

- $URI = \{URL, URN\}$
- URI: Unified Resource Identifier  
URL: Unified Resource Locator  
URN: Unified Resource Name  
(e.g., urn:isbn:0-13-020601-6)
- URL: {protocol/app., hostname, [port, directory, filename]}

# URL Examples

Proto	What it does	Example
file	accesses a local file	<code>file:///etc/syslog.conf</code>
ftp	accesses a remote file via FTP	<code><u>ftp://ftp.admin.com/adduser.tar.gz</u></code>
http	accesses a remote file via HTTP	<code><u>http://admin.com/index.html</u></code>
https	accesses a remote file via HTTP/SSL	<code><u>https://admin.com/order.shtml</u></code>
ldap	accesses LDAP directory services	<code><u>ldap://ldap.bigfoot.com:389/cn=Herb</u></code>
mailto	sends email to a designated address	<code><u>mailto:linux@book.admin.com</u></code>

# How HTTP works

- Stateless client/server protocol
- A client asks the server for the **contents** of a specific **URL**
- The server responds with the data (or err. msg.)
- Try it: telnet to port 80

# 人肉Browser

- telnet to www.csie.ntu.edu.tw port 80  
(http default TCP port)
- Type the following:  
GET / HTTP/1.1  
Host: www.csie.ntu.edu.tw  
(hit <enter> twice)
- What do you get?
- Try a nonexistent URL. What do you get?

# Content Generation

- Dynamic content is better
  1. CGI (Common Gateway Interface):  
Allow external program to interact with the web server
  2. FastCGI: Allow external program to continue running to server multiple requests
  3. Embedded interpreters:  
(e.g., Perl, PHP, Python, Ruby on Rails)  
Executing external script within the server (.php, .pl)  
e.g., LAMP: linux + apache + mysql + php/perl/python
  4. Application servers:  
Entire, full-fledge, platform for web  
(e.g., Tomcat, WebSphere, WebLogic, Jetty)

# Security!



- Bottom line:  
you allow **the entire world** to execute a script on your server (access to files, networks, and more!)
- Need to make sure that the script is secure (as much as other network-accessible program)



# Load Balancing

- Many factors affecting the maximum load a server can handle:
  - Hardware architecture
  - Operating system
  - System tuning
  - Sites being served  
(static vs dynamic (database))
  - And, in addition, network bandwidth
- Stress testing
- Implications in security

# Create Scalability

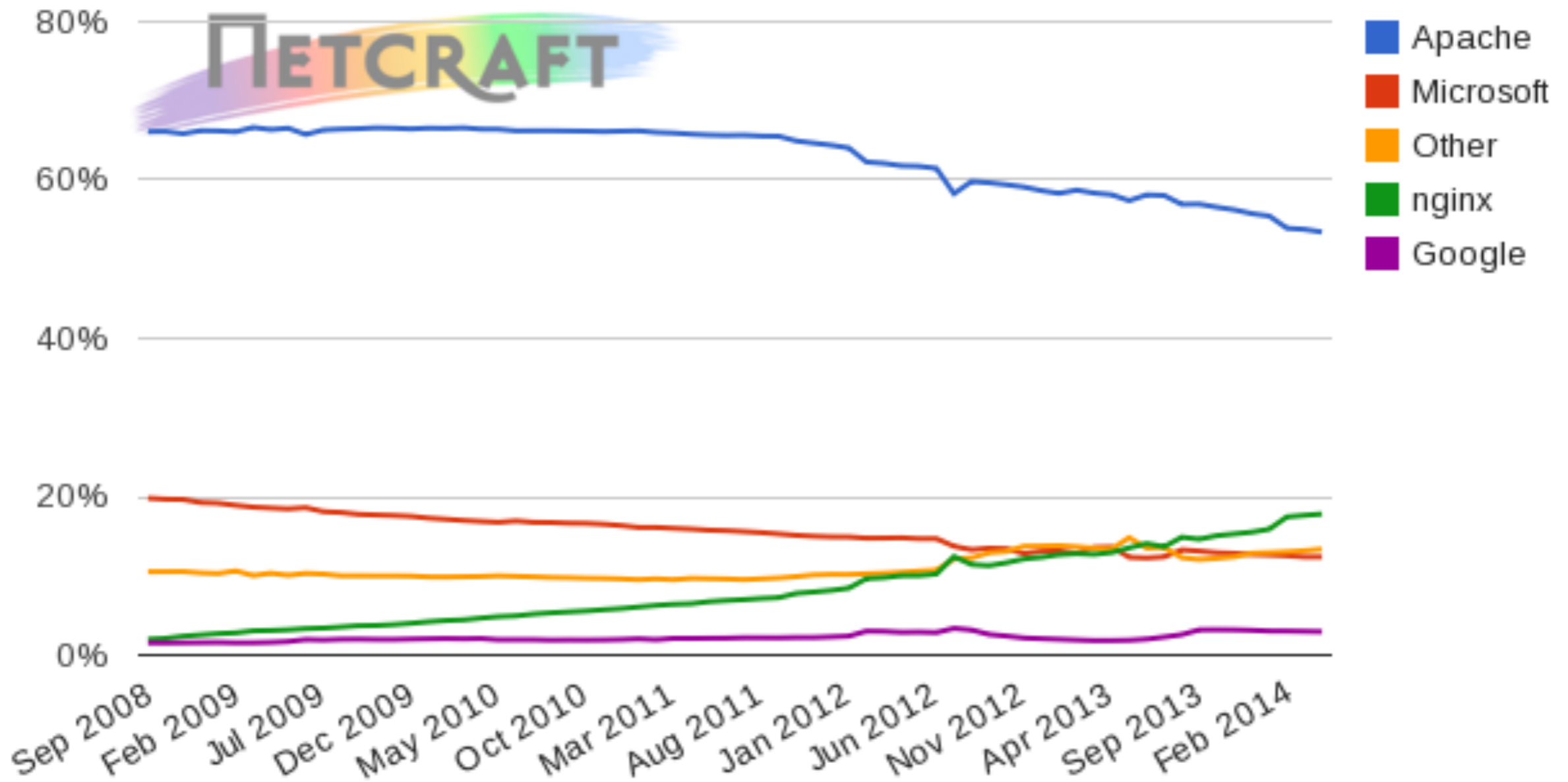
1. Round robin DNS (we've talked about it)  
Note that the order in the DNS record is **irrelevant**.  
(Think about its disadvantage)
2. Hardware solution (e.g., Big-IP from F5)  
Takes response time of individual servers into account
3. Software solution (e.g., Linux Virtual Server, proxy load balancing in Apache)

# Apache



- Web server with the largest market share (53.8% of top M busiest sites, 2014/04)
- Runner-ups:  
Microsoft & nginx take 17.8% and 12.4%
- Began in 1995
- First web server software to server more than 100M sites (in 2009)
- Versatile

Web server developers: Market share of the top million busiest sites



source: <http://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html>

# Hello World HTML

```
<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML>
  <HEAD>
    <TITLE>
      A Small Hello
    </TITLE>
  </HEAD>
  <BODY>
    <H1>Hi</H1>
    <P>This is very minimal "hello world" HTML document.</
P>
  </BODY>
</HTML>
```

You can copy the file to your user dir. for testing

# In-class Homework

- Install Apache in your VM
- Instead of serving via port 80, serve via port 8080
- Turn on UserDir module, and allow the users to publish their web pages at  
`http://<hostname>/~<user>/web/`  
(note that the default is `public_html`, not `web`, so you need to change the configuration)
- <Bonus> Install JMeter to stress test your web server  
Ref: <https://www.digitalocean.com/community/tutorials/how-to-use-apache-jmeter-to-perform-load-testing-on-a-web-server>  
**<<<DO NOT STRESS TEST ANY PRODUCTION SERVER!!!!>>>**

# Some Useful Tips



- Apache configuration files are in `/etc/apache2` (Ubuntu default installation)
- Create a symbolic link in `mods-enabled/` to linked to files in `mods-available/` to enable a certain module. (And if needed, modify the config files in `mods-available`)
- `apachectl` is the command line tool to issue command to the server  
e.g., `apachectl restart`