Today

- Review of today’s homework
- Web servers
- Web performance problems
- Web traffic measurement
- Break
- Student discussion topics
- WTC Attack: Akamai
- Future course topics
- For next week
Today’s Homework

- Q: Which RFC describes HTTP and what is its status?
  - A: Everyone who sent me a response said RFC 2616 (correct). One person found its status, which is that of a “Draft Standard”
  - Answers from W3C, various RFC archives, faqs.org, etc.

- Remember:
  - No exams does not mean no work. Earn your grade.
Web Servers

• Web site vs. Web server
  • What is a Web server?
    • Software that handles HTTP requests for particular resources.
  • What is a Web site?
    • Examples: university, e-commerce, organizations, portals, search, etc.
    • Load on site includes popularity, number and size of resources, dynamic content, access control, and connectivity of clients
Web Servers 2

• Handling a client request
  • Read and parse HTTP request message
  • Translate the URL to a file name
    • May be static, or part of a CGI
  • Determine whether the request is authorized
    • Perhaps by IP, or name and password
  • Generate and transmit the response
  • (write summary to log file)
Generating dynamic content

- Anything other than reading and sending a fixed file
- Many mechanisms
  - Server-side includes
    - Server parses HTML for extra tags
    - E.g., .shtml, .php, .asp
  - Server scripts/programs
    - Program generates entire response (HTTP headers + HTML content)
Executing server scripts

- As separate process (CGI)
- (Common Gateway Interface)
- As a module within the server
  - (Netscape’s NSAPI, Microsoft’s ISAPI, Apache’s mod_perl, Sun’s Java servlets)
- Persistent process contacted by server
  - (FastCGI, many other systems)
Advantages/Disadvantages?

- (of approaches to dynamic content generation)
- Performance, security
- Consider backend DB on separate machine, parsing scripts
Maintaining state between reqs

- Three possibilities
  - Authentication
    - Use username to lookup state
  - Cookies
    - Identifier to lookup (or store small) state
    - Set-Cookie: response header
    - Cookie: request header
  - Session ID in URL
    - Parameter to CGI
Performance optimizations

- Sharing information across requests
  - Server-side caching of responses
    - Eliminate open/read/close of static files
      - As opposed to implicit OS filesystem caching
    - Eliminate parse/calculate/lookup of dynamic files
  - Retain meta-data across requests
    - Translation of URL to file name
    - File system info (location, date, access)
    - HTTP response headers
    - Current date/time
    - Client name (IP lookup)
Server architectures

- Event-driven
  - Simplest: single process handling one request at a time to completion
  - Better: single process handles part of each request at a time, effectively serving many requests simultaneously

- Process-driven
  - Each request gets a separate process
  - OS switches automatically
  - Startup significant – pre-fork processes

- Hybrid
  - Examples: Apache, MS IIS, Flash, Zeus (HW: Classify all, find at least one more and classify)
Server hosting

- Single machine per site
- Multiple sites per machine
- Multiple machines per site
- Single IP or multiple IP addresses

How common are these approaches?
The Apache Web Server

- Most popular server on Web
- Process-based
- Parent spawns child processes in advance of request

(read more detail in chapter 4)
Web Performance Problems

• In general, speed/efficiency while maintaining correctness.
• From your list last time:
  • Too slow
  • Internet as a whole not immune to problems and failures
• How do we know such problems exist?
• Or, how will we know that we’ve made any improvements?
Web Traffic Measurement

• (from chapter 9)
• Sources of Web traffic data:
  • Web servers
  • Web proxies
  • Augmented browsers
  • Packet monitoring
  • Active measurement
Measurement Difficulties

- Caching (cache-busting?)
- Location of measurements
- What measurements are captured? Do they correspond to desired phenomena?
- Missing/incomplete/ambiguous/corrupted logs
Interlude

- Break
- Student discussion topics
WTC Attack:
Akamai Co-Founder Killed

- Daniel C. Lewin, 31, on Boston flight that crashed into WTC
- Co-founded Akamai, served as CTO
- A graduate student (PhD candidate) at MIT (masters thesis provided some core technologies licensed to Akamai)
Future course topics

- “How to speed the Web”
- Traffic measurement, analysis, and simulation
- TCP/IP, DNS
- HTTP
- Caching
- Load distribution – scaling for high demand
- Content Delivery Networks
- Speculative content transmission
- Peer-to-Peer content delivery
For Next Week

- Post at least one message on blackboard about something from one of the first two classes
  - E.g., a question, comment, or suggestion
- Read Chapters 5, 6, 9, 10 and parts of 14
- Classify Apache, MS IIS, Flash, Zeus plus one more
- As always, bring something interesting to discuss, relating to Web performance (weekly)