Web Services and Applications

Robert Grimm
New York University
- It’s all about the World Wide Web
- But, with a shift in focus
  - Away from human-centric web
    - People access information manually
  - Towards automated web
    - Computers access information automatically
An Example

- Shopping on the Internet
  - Directory web site
    - Find fitting item (say, digital camera)
    - Find best price
  - Vendor web site
    - Buy item (register, log in, select item, check out)
  - Shipping company web site
    - Track package (again and again)

- Automate process
  - Human selects item
  - Shopping agent does the rest
What’s Needed

- Some way to
  - Find services
    - Universal Description, Discovery, and Integration (UDDI)
  - Describe services
    - Web Service Description Language (WSDL)
  - Invoke services
    - XML-based Remote Procedure Calls (XML-RPC)
    - Simple Object Access Protocol (SOAP)
  - Communicate with services
    - HyperText Transport Protocol (HTTP)
Web Service Properties

- Internet accessible
- Standardized messaging
  - XML-based
  - Not tied to single OS or programming language
- Self-describing
- Discoverable
Course Goals

- Three goals
  - Understand web technologies
  - Hatch ideas for research
  - Develop methodology for building complex systems

- Focus on
  - Programming model
    - Also, data model
  - Engineering of large-scale systems
Course Components

- Three components
  - Readings to introduce topics
  - Class discussions to deepen understanding and develop new ideas
  - Programming assignments to provide hands-on experience

- Think
  - Research seminar
    - Learning by reflection and discussion
  - Systems building course
    - Learning by getting your hands dirty
Readings and Class
Readings and Class

- 2-3 required papers per week
  - Write one paragraph summary (per paper)
    - Main idea
    - Innovations (if any)
    - Your criticisms
    - Possible research directions
  - Participate in class discussion
    - I provide slides to review material and guide discussion
- Further readings for additional background
  - If interested, in your copious free time
Topics

- HTTP
  - The evolution of the web protocol
- Building fast servers
  - How to make a single server support a large population
- Clusters
  - How to use many off-the-shelf computers to support an even larger population
- Caching
  - How to avoid going all the way across the Internet
Topics (cont.)

- **Content: XML**
  - *The Internet data model*

- **Content: Multimedia**
  - Most of the real data

- **Small devices**
  - How to network resource-constrained devices

- **RPC**
  - How to invoke remote services

- **Security**
  - Worms and distributed denial of service attacks
• Descriptions
  • How to describe services

• Discovery
  • How to find services

• Active Everything
  • How to run code everywhere

• Representational State Transfer (REST) vs. SOAP
  • Are SOAP/WSDL/UDDI really the right approach?
What We Don’t Cover

- Content delivery networks (think Akamai)
- Peer-to-peer systems
- Data management systems
- Traditional security (think access control)
- Economics and Law
  - Micro-payments
  - FatWallet.com sued by Wal*Mart, Target, Best Buy, Staples, OfficeMax, Jo-Ann Stores, KMart
Programming Assignments
Programming Assignments

- Written in Java
  - Based on *Munin*, an event-driven web server
    - You are going to measure and extend it
- By groups of ~4 students
  - You are going to form groups today (yeah)!
  - First assignment will be posted next week and be due two weeks thereafter
Three More (Required) Tasks

- Test for interoperability
- Track your efforts
- Document your results
Perform Interoperability Testing

- This class focuses on standardized protocols
  - Testing your group’s client with your group’s server is not enough
  - But don’t want to unleash (potentially) buggy code onto the Internet at large
- Therefore
  - Each group needs to test its client with another group’s server and vice versa
Track Your Efforts

- Time in hours per person spent on
  - Preparation
  - Design
  - Implementation
  - Basic testing and debugging
  - Interoperability testing
  - Documentation and write-up
- Lines of code (use JavaNCSS)
- Number of bugs
  - When introduced and when fixed
Document Your Results

- A 3-5 page extended abstract per assignment
- Provide an overview of your goals and design
- Convince me
  - Your client and server work
    - Comply with standards, handle error cases correctly
  - Your client and server meet your design goals
    - Perform a certain way
  - Other group’s client and server work
- Report your project’s statistics
- Share interesting anecdotes (e.g., what was surprising)
Some Application Ideas
(For Your Final Assignment)

- Discussion board
  - Think: Paper summaries
- Time tracker
  - Think: Productivity tracking
- Web cam proxy
  - Think: George Orwell or JenCam
- Visitor announcement and tracking
  - Look at 7th floor lobbies at 715 Broadway
Our Web Cams

- Axis web cams with embedded web server
  - Directly connected to Internet
  - Support only up to 10 simultaneous connections
- One camera overlooking Manhattan
  - http://66.93.85.13/
- Two cameras on the 7th floor at 715 Broadway
  - orwell1.cs.nyu.edu, orwell2.cs.nyu.edu
A Few More Things
Collaboration Policy

- Discuss readings and topics with each other
- But write reading summaries individually
- Help each other with programming questions and interoperability testing
- But do not share code outside groups
  - No sharing between groups
  - No code from outside resources (such as Internet)
  - No java.net.URL, java.net.URLConnection
One web site
   - http://www.cs.nyu.edu/rgrimm/teaching/sp03-web/

Two mailing lists
   - g22_3033_008_fa03@cs.nyu.edu
   - g22_3033_008_fa03-readings@cs.nyu.edu
   - Subscribe to both lists
   - Post only plain-text messages, not HTML

x groups
   - Start forming groups now, notify me by Friday
No official office hours
  - Just drop by
  - Email me to schedule a meeting

- 715 Broadway, room 711
Let’s Get Started: HTTP/1.0
HTTP/1.0

- Follows request/response model
  - Each request from client to server receives a response from server to client
    - Request: Client → Server
    - Response: Client ← Server

- Layered on top of TCP
  - One connection per request/response interaction

- Performs methods on resources
  - Resources named by URLs
  - Methods are GET, HEAD, and POST
The Three HTTP/1.0 Methods

- **GET**
  - Retrieve the contents of a resource

- **HEAD**
  - Just like GET, but the contents are not returned

- **POST**
  - Add data to a resource
    - Don’t implement without authentication!
Structure of Requests and Responses

- Initial request/response line
  - GET /path/to/file/index.html HTTP/1.0
  - HTTP/1.0 200 OK

- Additional header lines
  - User-Agent: Mozilla

- Optional message body
  - Form data
  - HTML page
The Client

- Opens TCP connection to server (usually port 80)
- Sends request
- Reads response
- Processes response
  - E.g., displays resource contents to user
The Server

- Listens for TCP connections
  - Accepts client connections
- Reads request
- Processes request
  - E.g., reads file
- Sends response
- Closes TCP connection
Composing Clients and Servers: The Proxy Principle

- Interpose on a client/server interaction
  - Without proxy: Client ↔ Server
  - With proxy: Client ↔ (Server+Client) ↔ Server

- Enable
  - Transformation
  - Aggregation
  - Caching
  - Customization
Some Design Trade-Offs

- **Performance and scalability**
  - Event-based servers tend to perform/scale better
    - Java 1.4 supports asynchronous I/O (java.nio)
    - See Flash and SEDA papers from syllabus

- **Resource consumption**
  - Memory, number of threads, CPU consumption under increasing load

- **Extensibility and configurability**
  - Modularity of server
Forgiveness

- Be liberal in what you accept, be conservative in what you send
  - E.g., always terminate lines with CRLF but accept lines only terminated with LF

- But, many denial of service attacks exploit this credo
  - ///////////////////////////////////////////////////////////////////////////
  - User-Agent: Slow sender
Finally: Meet and Greet