COMP 117: Internet-scale Distributed Systems

Lessons from the World Wide Web

Noah Mendelsohn
Tufts University
Email: noah@cs.tufts.edu
Web: http://www.cs.tufts.edu/~noah
What you should get from today’s session

- You know what course is about and why you would want to be here
- You know who I am
- You know mechanics of the course and what will be expected of you
- If there’s time...start on some technical work
Introduction
The course will explore several threads in parallel:

1. We will look top-down at various principles of distributed system architecture
2. We learn to build Internet clients and servers that communicate
3. We will contrast several styles of distributed systems
4. We will explore the history of computer networking, and explain why the Web is designed the way it is
5. We will look at a variety of other related topics
What this course is not

- A course on designing Web applications (COMP 20/120)
- An comprehensive OS course (COMP 111)
- An comprehensive networking course (COMP 112)
- A programming languages course (several)
What this course is not

- A course on designing Web applications (COMP 20/120)
- An comprehensive OS course (COMP 111)
- An comprehensive networking course (COMP 112)
- A programming languages course (several)
- However...we will explore issues relating to all of those!

Prerequisites

- **Formal requirement**
  - COMP 40 or permission
- **Informal requirement**
  - Good ability to program in C/C++ at level of COMP 11/15 (structs, classes, methods assumed; we will teach here but not assume knowledge of inheritance and exceptions)
  - Ability to deal with complex systems challenges at level of COMP 40
  - Assume you know how to use Hallgan facilities, ssh into linux.eecs.tufts.edu, etc.
Noah Mendelsohn – some background

- 40+ years building & teaching distributed systems
  - Mainly at IBM and Lotus
  - Lots of time working w/ university research groups (Stanford, MIT, UCLA)
  - Worked mainly at intersection of research and advanced technology development
  - Have worked on many important and interesting systems
  - Taught programming at Stanford
  - Known for contributions to Java, XML, Web, etc.
  - Thru spring 2013: chair of the World Wide Web Consortium TAG

- Current
  - Professor of the Practice in Computer Science at Tufts University – since 2012
  - IBM Distinguished Engineer Emeritus

- Contact and additional info:
  - noah@cs.tufts.edu
  - http://www.eecs.tufts.edu/~noah/ (more available from there)
  - http://www.arcanedomain.com (personal Web site – see esp. list of recommended CS readings)

It’s really fine to call me Noah. If you prefer, Prof. Mendelsohn is also OK.
COMP 117:
Internet-scale Distributed Systems –
Lessons from the World Wide Web
COMP 117: Internet-scale Distributed Systems
*Lessons from the World Wide Web*
Tim Berners-Lee

Inventor of the World Wide Web
Director of the World Wide Web Consortium (W3C)
The World Wide Web Consortium (W3C)
What the W3C does*

- Create and promote the standards that make the Web work
- Work to ensure the openness of the Web
- Etc.

* Another organization called the IETF also plays a key role in creating core Internet standards
Why Document the Principles of Web Architecture?

These statements of architectural principle explain the thinking behind the specifications. [...] They are aimed at the technical community, to explain reasons, provide a framework to provide consistency for future developments, and avoid repetition of discussions once resolved.
[Tim Berners-Lee October 1998 ]

From: http://www.w3.org/DesignIssues/Preface.html
The World Wide Web Consortium (W3C) Technical Architecture Group (TAG)
The W3C Technical Architecture Group (TAG) - 2012

The senior technical body responsible for the Web
What the TAG does

- Document the architecture of the Web
- Resolve issues raised by members of the Web community or W3C workgroups
- Coordinate cross-technology architecture developments inside and outside W3C
Useful publications of the TAG

- Architecture of the World Wide Web (Vol. 1 – there is no Vol. 2!)
  http://www.w3.org/TR/webarch/

- TAG “Findings” on specific architectural topics:
  http://www.w3.org/2001/tag/findings

- We will read some of these during the course – you are encouraged to read more on your own
References from Tim Berners-Lee

- Tim BL’s home page: [http://www.w3.org/People/Berners-Lee/](http://www.w3.org/People/Berners-Lee/)
- Tim BL’s design notes on the Web: [http://www.w3.org/DesignIssues/](http://www.w3.org/DesignIssues/)
- Web Architecture from 50,000 feet (Tim BL): [http://www.w3.org/DesignIssues/Axioms.html#Universality](http://www.w3.org/DesignIssues/Axioms.html#Universality)

We will explore many of these references in detail later in the course. For now, you might want to take a look if you’re curious.
Course Plan
Course Web Site

http://www.cs.tufts.edu/comp/150IDS/

Course home  Piazza
Syllabus/Lectures  Assignments  Info

**Principles**
Assignments & Tests

- Multiple assignments will sometimes run in parallel – manage your time!
- Reading Q&A:
  - Most reading assignments will come with questions.
  - Two due dates: 1) in time for class discussion, mostly ungraded 2) same questions after discussion, graded
  - Learning to do technical analyses and communicate your insights is important!
  - For most assignments, you will “provide” answers by filling out and submitting an HTML form that will be supplied to you.
- There will be several distributed programming assignments.
  - You must ssh into virtual servers at Halligan to test network programs
  - We will use team programming for larger projects
- Tests
  - One or two tests (typically announced in advance) + in-class final exam
- Final project
  - There will be a choice of projects, and you may propose your own. The typical project will be in the form of a paper analyzing some important topic relating to the course. Final projects may involve code, but not necessarily. It will depend on your choice of topic.
  - Assigned later in term, due during finals week
Grade scales and curving

- **Course grading:**
  - Just do the work that’s asked, you’ll probably get about a B
  - Do it with extra style and care, or show some extra creativity, that’s an A
  - Do less than a good job, that’s B- or below

- **Assignment grading:** modeled on COMP-40 but using numbers
  - Scale: 0-15
  - 12 is roughly an A, 9 is roughly B, 6 is roughly C
  - > 12 is excellent, A+, over-the-top (and rare) – this scale is designed to allow you to get credit for excellence
  - …DO NOT PANIC IF YOU GET A 9 OUT OF 15, IT REALLY, REALLY IS ROUGHLY A B!

- **Curving**
  - I will not curve grades with the intention of ensuring a fixed distribution of A’s, B’s, C’s – my hope is that you will all earn A’s, but that does mean you could all get Bs or Cs.
  - I may curve results up or down if I feel that a test or assignment did not accurately measure what I intended. I may also just decide not to count such work if it would lower your grade.

- **Coding**
  - Coding “standards” will be suggested – not in all cases the same as for COMP 40
  - Mainly: your code must be well structured, modular, and easy to understand
  - The source of your code must be visually appealing and easy to navigate
  - Use as many well-formatted comments as necessary to make your code comprehensible: a comment is good if the code is more understandable with it than without
Your final course grade

- **Grade weighting**
  - Current plan is to weight: (20% in class participation and reading q&a; 30% programming projects, 30% tests & final exam, 20% final project). I’ll announce final weightings later.

- **If you do all your work on time, you are guaranteed a grade no worse than this.**

- **You can do better than this by getting extra credit for work that goes “above and beyond”**
  - Some assignments will offer extra credit topics/questions
  - If you have ideas for extra credit work or alternate assignments, ask me
  - You do not need to do extra credit work to get an A, but you do need to do a thorough and creative job on the assignments. You should try for extra credit if you’re excited about the material, or have extra insights that you’d like to share.

- **You may lose credit if your work is late or you miss class too often**

- **All extra credit, and all penalties for missing or late work are at my discretion**
  - I will try to be fair, I understand that people get sick or have other work, and I have no interest in tripping you up over trivialities. That said, late work puts a strain on me and the TAs, and sometimes we will be discussing the “answers” in class. Extra or lost credit may or may not affect your grade (typically depending on how close you are to a different grade, and how much extra/lost you have).

- **I may raise (but not lower!) your final grade**
  - …if I feel that your overall knowledge of the material has been demonstrated – e.g. in discussions with me

- **There will be no incompletes given. Exceptions will be made only in extreme circumstances after consultation with the dean.**

In short... the computed grade is a lower bound if all your work is on time!
Piazza

- **Piazza is an online discussion and course management system**

- **We will use it for:**
  - Answering your questions
  - Announcements from the course staff

- **Discussion and asking questions**
  - We prefer if you ask questions in public on Piazza
  - You can help each other with answers
  - Everyone can see what’s causing confusion
  - Obviously: don’t post answers to hw questions or fragments of solution code
  - You can discuss such things in “instructor-only” Piazza posts
  - If you have a truly private matter email ta117ids@cs.tufts.edu or noah@cs.tufts.edu

- **Sign up for Piazza now!**
Reading Materials

- There is no textbook for this course
  - *There are no books you must buy*
  - You will be assigned selected readings from several books, *(almost) all of which are available on Safari books online.* Information on accessing these is available from the course Web site.
  - You *may* want to consider getting one or more of:
    - Kerrisk, M. *The Linux Programming Interface* – a terrific book covering all the Linux APIs + networking concepts
    - Whatever book suits you for learning advanced C++ topics. I use the book by the inventor of C++: Stroustrup, B. *The C++ Programming Language* but you may prefer another.
    - Nemeth, et. al., *Unix and Linux System Administration Handbook (4th Ed.)* – we’re not learning system admin, but this has very comprehensive information about Unix/Linux command line programming
    - Kerrisk, Nemeth and Stroustrup are all on Safari, and many alternative books on these topics are too
  - Berners-Lee, Tim, *Weaving the Web*
Reading Materials

- There is no textbook for this course
  - *There are no books you must buy*
  - You will be assigned selected readings from several books, *(almost) all of which are available on Safari books online.* Information on accessing these is available from the course Web site.
  - You *may* want to consider getting one or more of:
    - Kerrisk, M. *The Linux Programming Interface* – a terrific book covering all the Linux APIs + networking concepts
    - Whatever book suits you for learning advanced C++ topics. I use the book by the inventor of C++: Stroustrup, B. *The C++ Programming Language* but you may prefer another.
    - Nemeth, et. al., *Unix and Linux System Administration Handbook (4th Ed.)* – we’re not learning system admin, but this has very comprehensive information about Unix/Linux command line programming
    - Kerrisk, Nemeth and Stroustrup are all on Safari, and many alternative books on these topics are too
  - Berners-Lee, Tim, *Weaving the Web* (shared copies available compliments of Tim)

- I will assign selected other readings. All necessary papers will be available online.

- When I use slide sets like this, they will be posted online in .ppt and .pdf formats (please bug me if I forget!)

- Where published material or slides does not provide sufficient coverage, I will prepare course notes
Administrivia

- Noah’s office hours: Tues/Thurs after class, or by appt.
- Best way to reach Noah quickly: noah@cs.tufts.edu.
- For near emergencies: phone or text 617-506-3994 (Google voice)
- TAs:
  - Ethan Pailes
  - Kabir Singh
  - Email: ta117ids@cs.tufts.edu
- Remember: most questions should be asked on Piazza
Homework Assignments Now Posted
Three homework assignments being given today:

1. Due Tues. Jan. 24: E-mail to instructor
   (http://www.cs.tufts.edu/comp/150IDS/assts/emailassignment)

2. Due Tues. Jan. 24: Sign up for Piazza
   (https://piazza.com/tufts/spring2016/comp150ids/home)

3. Due Feb. 7 if you are in first group to receive book: Read chapters from Weaving the Web + 2001 Time Magazine article and answer questions:
   (http://www.cs.tufts.edu/comp/150IDS/assts/weavingtheweb)

There will be more assignments given early next week that overlap with the reading assignment. Plan accordingly (allowances will be made for those who get the Tim BL book later)
Questions?
Credits

- The picture of Falling Water shown on slide #2 is from Happy Via’s’ photostream on Flickr. (http://www.flickr.com/photos/via/57100237/ License: http://creativecommons.org/licenses/by-nc-nd/2.0/)