The cloud and the edge

Image from http://www.vs.inf.ethz.ch/about/zeit.jpg

Today

- Course introduction
- Distributed systems and their challenges

Next time

- Systems models and LOCKSS

Welcome to EECS 345!

- What's all about?
  - A course on the edge, the cloud and related systems technology under all
  - A lecture/seminar hybrid with lectures and paper discussion, project and exam, …
- Our goals
  - Learn about distributed systems,
  - … basic principles and current research
  - Learn/practice reading research papers,
  - … arguing and convey ideas, and
  - … generating a research project of your own

What is a *distributed system*?

- Very broad definition
  - A collection of independent, interconnected processors that communicate and coordinate their action by exchanging messages
  - A collection of independent computers that appears to its users as a single coherent system
- Why do you want one?
  - Resource sharing – both, physical resources and **information**
  - Computation speedup – to solve large problems, we will need many cooperating machines
  - Reliability – machines fail frequently
  - Communication – people collaborating from remote sites
  - Many applications are by their nature distributed (ATMs, airline ticket reservation, etc)

Loosely, closely, tightly

- Most distributed systems are “loosely-coupled”
- Each system is a completely autonomous system, connected to others on the network
- Earliest systems
  - FTP (rcp): file transfer program
  - telnet (rlogin/rsh): remote login program
  - mail (SMTP)
- Most distributed systems are loosely-coupled
  - Each CPU runs an independent autonomous OS
  - Computers don’t really trust each other
  - Some resources are shared, but most are not
  - The system may look differently from different hosts
  - Typically, communication times are long

Closely-coupled systems

- A DS becomes more “closely-coupled” as it
  - Appears more uniform in nature
  - Runs a “single” operating system
  - Has a single security domain
  - Shares all logical resources (e.g., files)
  - Shares all physical resources (CPUs, memory, disks, printers, etc.)
  - A previous version of your textbook was called “Distributed Operating Systems”
- In the limit, a distributed system looks to users as a centralized timesharing system, but built of a distributed set of hardware and software components

Tightly-coupled systems

- A “tightly-coupled” system usually refers to a multiprocessor
  - Runs a single copy of the OS with a single job queue
  - Has a single address space
  - Usually has a single bus or backplane to which all processors and memories are connected
  - Has very low communication latency
  - Processors communicate through shared memory
- SOSP 2009 paper Baumann et al., “The Multikernel: A new OS architecture for scalable multicore systems”

  “We argue that the challenge of future multicore hardware is best met by embracing the networked nature of the machine, ***rethinking OS architecture using ideas from distributed systems****”*

Distributed systems challenges

- Making resources available
  - The main goal of DS – making convenient to share resources
- Security
  - Sharing, as always, introduces security issues
- Providing transparency
  - Hide the fact that the system **is** distributed
  - Types of transparency
    - Access – What's data representation?
    - Location – Where's the resource located?
    - Replication – Are there multiple copies?
    - Concurrency – Is there anybody else accessing the resource now?
    - Failure – Has it been working all along?
    - …
  - Do we **really** want transparency?

Distributed systems challenges

- Openness
  - Services should follow agreed-upon rules on component syntax & semantics
- Scalability
  - In numbers (users and resources), geographic span and administration complexity
  - Some useful techniques
    - Asynchronous communication
    - Distribution
    - Caching/replication

Distributed systems challenges

- Adding to the challenges, common false assumptions
  - The network is reliable
  - The network is secure
  - The network is homogenous
  - The topology does not change
  - Latency is zero
  - Bandwidth is infinite
  - Transport cost is zero
  - There is one administrator

Cloud & edge – Side-by-side revolutions

- Edge computing – direct interactions among computers (peers) out in the Internet
  - In 2003 SETI@Home beat the top 4 of the top500 supercomputers
  - P2P file sharing may generate about 1/3 of all Internet traffic
- Cloud computing – Move computing functions into large shared data centers
  - Amazon “hosts” data centers for customers
  - Google runs all sorts of office apps
- What’s the best place for …?
  - Storing massive amounts of data
  - Running scalable services
  - Connecting people
  - …

EECS 345

- Clearly we cannot cover all topics in distributed systems in one quarter
- Instead, we will
  - Introduce principles of distributed systems
  - Review ideas and ongoing work to ground our discussion
  - Following a new seminar/lecture model

Typical seminar model and its problems

- Our own old model
  - Read and discuss papers
  - Pick a project early on (or we'll give you some)
  - Work on the project to potentially have the seed of a paper the end of the quarter
- Problems – you may …
  - Not be sure about a topic
  - Not know about the related work
  - Not have a good idea on time to write a proposal
  - Simple run out of time
  - …

EECS 345 new model

- Get a good grasp of the basic ideas before discussing papers
- Learn a bit about background work as to generate one cool, potentially publishable idea per group
  - Take it up to a proposal stage
  - What you do with it after the quarter is over is up to you! Independent study?
- Still work on a small project to get some hands-on experience and learn the basics of distributed systems

Grading

- Class 30%
  - Paper reading summary and presentation 20%
  - Class participation and discussion 10%
- Project 25%
  - A small focused project to give you some experience on the basics
- Research idea 20%
  - Presented as a research proposal, about 3 pages long
  - **The proposal document will be your only deliverable for this part**
- Take-home exam 20%

Paper summary

- For every paper – a one-page summary **due by 11:59PM of the previous day**
  - Brief one-line summary
  - A paragraph of the most important ideas.
  - A paragraph of the largest flaws – Being able to assess weaknesses/strengths is an important skill
  - A paragraph stating relevance of the ideas today, potential future research suggested by the article, etc.
  - Don't just restate the abstract/conclusion of the paper!
  - You can miss up to one paper summary w/o consequences
- Useful references (both available in the course site)
  - *Efficient reading of papers in Science and Technology*
  - Redell, *An evaluation of the ninth SOSP submissions or How (and how not) to write a good systems paper,*

Paper presentation

- Plan for a 20-25’ presentation; typical for a conference
- Keep in mind
  - A presentation is a performance, carefully planned & rehearsed
  - You are the main advocate of the work, act as such
  - Respect your audience – do not waste y/our time with filler
  - Make slides visually pleasant w/o a Madison Avenue look
- Slides will be posted in the course website

  (with your name next to it!)

Project

- Idea
  - Basic introductory project
  - Up to two-people team
  - First half of the quarter
- Possible topics
  - Write a small web server
  - Setup a test your own DNS server
  - Out next Monday, due around midterm
- Deliverables
  - Code and demo
  - Write up

Project proposal

- Successful projects start with clearly stated goals
- Idea – Get a project started by writing a proposal for it
  - Project startup document following John Wilkes' guidelines
- Basic idea: think of the project as a hypothesis-experiment-conclusion chain
- Basic sections
  - Problem statement
  - Proposal
  - Hypothesis
  - Experiments
  - Results

Content in a Startup Document

- Problem statement
  - What's the problem? Why does it matter? Who cares?
- Proposal
  - What is the basic approach, method, idea or tool being suggested to solve it?
- Hypotheses
  - Expected effects of the proposed solution? Why?
  - Plausible alternatives & how likely are they?
  - What's good/bad about them by comparison?
  - What have others done already? What did they learn?

Content in a Startup Document

- Experiments
  - What will be done to test out the hypotheses?
  - How will this confirm/deny it?
  - Why will the conclusion be believable?
- Results
  - What will be the outcome of the work?
  - When? Intermediate milestones?
  - How will we know when they are complete?
  - What are the measures for success?

Take-home exam

- Open-book (yes, that means web access ☺)
- About five in-depth questions
- I'll be making the following assumptions:
  - You have read all papers listed
  - You have acquired the required background
  - You understand the issues at play
  - You can critically read a paper

Next time

- An overview of major systems models using a couple of current papers to make our discussion more concrete
  - Paper: Maniatis et al., The LOCKSS Peer-to-Peer Digital Preservation System
  - Presenter: Fabián
- TODO
  - Go to the course site, pick a couple of papers you think you may want to present and email their titles to me