Curriculum Workshop Report

Jeannie Albrecht

Williams College

http://www.cs.williams.edu/~jeannie/nsf-workshop/index.html
http://groups.geni.net/geni/wiki/GEC14Agenda/Curriculum

Challenges

- Distributed systems provide many advantages
  - Improved scalability, fault tolerance, response time, etc
  - Widely used by many major companies
- Developing, debugging, deploying distributed systems introduces new challenges
  - Can be overwhelming to new developers
- Students benefit from the opportunity to design, implement, deploy, and evaluate ***real*** systems in ***real*** environments
  - Learn techniques for coping with common challenges

Overcoming Limitations

- Unfortunately undergraduate curriculums rarely offer Distributed Systems courses
  - Students are not fully prepared for jobs/grad school
- Many small colleges and some universities do not have the local computing resources required for large-scale experimentation
- Options are mainly small-scale emulation and simulation, which have advantages, but are not always realistic

Advancements in Technology

- New options have appeared in last decade
- There are now many public testbeds available
  - Developers can "rent" Amazon, Google, and Microsoft resources
  - Shared platforms like GENI provide a variety of deployment options
- Role of MOOCs (Massively Open Online Courses)

Workshop Goals

- (Re)Define undergraduate distributed systems educational goals
- Discuss ways to integrate new technologies with classic concepts
- Leverage availability of public testbeds to give students at a variety of institutions hands-on experience with the development and analysis of distributed systems

Session 1 – Platforms

- Getting access, managing student accounts, how to use in classroom, etc.
  - **Justin Cappos** (NYU Poly)
    - Seattle platform and testbed
  - **Gary Wong** (Univ of Utah)
    - Emulab and ProtoGENI
  - **Armando Fox** (UC Berkeley)
    - Amazon EC2 and other cloud-based technology
  - **Jeff Chase** (Duke Univ)
    - ExoGENI and ORCA

Session 2 – Educator Experiences

- Assignments, textbooks, etc., primarily used in small colleges and universities (undergraduate level)
  - **Jeannie Albrecht** (Williams College)
    - Distributed Systems
  - **Joel Sommers** (Colgate University)
    - Computer Networks
  - **Tia Newhall** (Swarthmore College)
    - Parallel & Dist Computing
  - **Zongming Fei** (Univ of Kentucky)
    - Networks & Dist OS/Calvin College Emulab

Session 3 – Educator Experiences

- Assignments, textbooks, etc., primarily used in large universities (ugrad and grad level)
  - **Armando Fox** (UC Berkeley)
    - Massively open online courses (MOOCs)
  - **Sonia Fahmy / Ethan Blanton** (Purdue)
    - GENI-based classroom exercises
  - **Anish Arora** (The Ohio State)
    - Projects designed for local/remote sensor testbeds/kits
  - **Mark Berman** (BBN/GENI)
    - Sample assignments designed for GENI

Common Themes: Learning Strategies

- Favor breadth over depth
  - Introduce students to a variety of technologies
- "Learn by doing"
  - Give students hands-on experience with deploying, debugging, analyzing distributed systems
- Development is not enough
  - Teach students about experimentation/evaluation/analysis
- Understand low level details
  - Students need to understand how things work at the socket level (at least)
  - Higher level languages (i.e., Java, Python, etc) greatly simplify development, but hide some details

Common Themes: Course Structure

- Textbooks provide good background knowledge, but students also benefit from reading research papers
- Emphasize scientific/technical writing in addition to code development
- Short/well-defined labs and assignments early in semester help students gain necessary background
- Open-ended/independent final projects allow students to explore and innovate on topics that they find interesting

Common Themes: Integrating New Technology

- Giving students experience with heterogeneous/ wide-area network conditions is important
  - But debugging in wide-area is challenging and frustrating, especially for students
  - Students have tendency to (often incorrectly) blame testbed when things don't work
- Important to let students work with a variety of resources in different environments
  - External resources (i.e., GENI) are very valuable for small colleges *and* large universities

Common Themes: Skills and Concepts

- Specific things students should know:
  - How to use a Makefile
  - Revision control
  - Debugging tools (gdb, valgrind, etc)
  - Scripting
  - Latex
  - Gnuplot/scriptable graphing software
  - Threads
  - Sockets

Conclusions

- Hard to reach consensus (at this point) on how undergrad Dist Sys course should be structured
  - Student backgrounds vary widely (i.e., different prerequisites, different levels of competency, etc.)
  - Some courses cover Parallel and Distributed Computing, some cover Networks and Distributed Systems, etc.
- Goal: Identify a set of modules/assignments that can be integrated into undergraduate courses
  - Allow instructors some personal freedom while also providing some curricular consistency
  - Low overhead assignments that can be easily inserted into existing curriculum

Conclusions

- Building real systems on real (wide-area) networks are important skills for students (especially undergraduates)
  - Testbeds (i.e., GENI) are critically important to achieving this!
  - …but GENI is still intimidating to educators (simplicity is key!)
- What can we do to make GENI less scary?
  - Minimize adoption hurdles – need "tried and true" tutorials/ modules for both educators and students
  - Minimize management overhead for instructors (we are generally pretty busy) and minimize student frustration/ confusion
- Reach out to educators at non-GENI events