Using GENI to Bring BIG Systems to small Schools

Jeannie Albrecht
Williams College
Introduction

• Undergraduates enrolled in a Distributed Systems course at Williams use GENI to gain hands-on experience with computer networks and “big” systems

• Goal: Teach students how to design, implement, and evaluate distributed systems

• Without computing platforms like GENI, students at small colleges lack the computing infrastructure necessary to deploy and evaluate distributed systems
Williams College

• About Williams
  • Liberal arts college in rural western Massachusetts
  • 2200 undergraduate students (no grad students)
  • Student:faculty ratio is 7:1

• CS@Williams
  • Avg: 15 majors per year (~3 women)
  • This year: 38 majors in junior class (12 women)
  • Many students double major
  • ~1/3 of our students go on to top tier graduate programs
  • 8 CS faculty members
  • Class sizes range from 30 in intro courses to 10-20 in upper-level electives (though this will likely increase!)
Course Overview

• **Goals**
  • Introduce students to key design principles
  • Teach students skills necessary to build and evaluate distributed systems
  • Expose students to cutting-edge real-world technologies
  • Improve technical writing skills

• **Components**
  • Programming projects (x4)
  • Midterm and final exam
  • Research paper evaluations (x8-10)
Student Profile

- **Prerequisites**
  - Data Structures
  - Computer Organization

- **Non-prerequisites**
  - Networks
  - Operating Systems

- First “project” course for many students

- Sample class breakdown
  - S08: 14 students: 2 sophomores, 4 juniors, 8 seniors
  - S12: 15 students: 1 sophomore, 6 juniors, 9 seniors
Project Overview

- Projects are 45% of overall grade
- Students work alone or with a partner
- Projects designed to emphasize techniques and technology from lecture topics and reading assignments
- Projects include a technical writing component
- Explored four different architectural models: client-server, multi-tier client-server, cluster computing, wide-area computing
Project 1: Web Server

• Assignment: Build a web server (in C)
  • Support GET requests in HTTP1.0 and HTTP1.1
  • Return valid response codes
  • Time allowed: ~2.5 weeks

• Goals
  • Explore simple client-server distributed computing paradigm
  • Gain experience with network/socket programming
  • Evaluate performance of HTTP1.0 and HTTP1.1 under varying conditions—hard to do using only local resources!

• Role of GENI (S14)
  • Create topologies (rspecs) with varying network conditions
  • Much like Hello GENI Example!
Project 2: Online Bookstore

- Assignment: Build a multi-tier online bookstore with “proper” synchronization
  - Use Java/Python and Java RMI XML-RPC
  - Timeline: ~2 weeks
- Goals
  - Explore multi-tier distributed computing paradigm
  - Gain experience with RPCs
  - Evaluate performance under varying levels of (artificial) load
- Role of GENI (S14)
  - Provide varying network conditions
  - (Same as webserver)
Project 3 v1: Inverted Index

• Assignment: Build an inverted index using Hadoop
  • Return valid mapping of words to documents using eBooks from Project Gutenberg as input
  • Timeline: ~3 weeks

• Setup
  • Created 60+ Xen virtual machines to host Hadoop mini-clusters using 14 cluster machines at Williams
  • Students maintained/configured their own cluster

• Goals
  • Explore “cutting-edge” cluster computing paradigm
  • Gain experience with basic system administration (without getting overly frustrated)
Project 3 v2: Contextual Advertising

• Assignment: Given an advertising context, predict which ad is most likely to be clicked (using Hadoop)
  • Compute click-through rate for ad id and page URL
  • Timeline: ~3 weeks

• Setup
  • Created small clusters on Amazon EC2 platform
  • Dataset also comes from Amazon
  • Students maintained/configured their own cluster

• Goals
  • Explore “cutting-edge” cluster computing paradigm

• Role of GENI – wide-area Hadoop??
Project 4 v1: P2P Computing

• Assignment: Build a P2P system (file sharing, game, distributed hash table, etc.)
  • Run system on PlanetLab
  • Be creative with design and implementation of system
  • Timeline: ~3.5 weeks (with strict checkpoints)

• Setup
  • Created each group their own PlanetLab slice
  • Students used Plush/Gush for app management

• Goals
  • Explore P2P wide-area distributed computing paradigm
  • Allow students freedom to innovate
Project 4 v2: Final Project

- Assignment: Open-ended final project
- “Default” project: Build a P2P file-sharing system
  - Run system on GENI
- Setup
  - Created each group their own GENI slice
  - Students used Gush for app management (PL and Emulab)
- Goals
  - Allow students freedom to innovate
  - Experiment with wide-area deployment
- Student results
  - Up to 400 GENI resources used
Gush User Interfaces

- Command-line interface used to interact with applications
- Nebula (GUI) allows users to describe, run, & visualize applications
- XML-RPC interface for managing applications programmatically
Student Feedback

• “[The final project] was one of the hardest and most rewarding projects I’ve done at Williams.”

• “I really felt like this was one of the most real-life applicable CSCI courses I took at Williams.”

• “I loved the papers! This was the first class that required critical responses to papers like that and I was surprised by how much I enjoyed it.”

• “Evaluating the papers, while kind of a pain sometimes, was actually quite valuable in retrospect; I learned a lot about distributed systems that way, and I’m glad we did them.”
Instructor Feedback

• Students really love Projects 1 and 4 (three students turned Project 4 into senior theses)
  • Some students appreciate open-endedness of Project 4; some struggle with it (default project helps)
• I spend (at least) 4-5 hrs per wk in lab helping students
  • Students work an avg of 10 hours per week
• Students miss the point of evaluation in early projects when only using local resources
  • GENI will help!
  • Perhaps introduce GENI experimenter tools (Gush, Flack, etc) early in semester rather than waiting until final project
• Good writers != good technical writers
Conclusions

• We should teach undergraduates how to design, implement, and evaluate real distributed systems
• Shared computing platforms (like GENI and EC2) provide students with the opportunity to gain hands-on experience with large-scale, wide-area distributed computing environments
  • Use shared platforms as learning laboratories
  • Bring tech-richness of big universities to small colleges
• Frameworks and tools like Hadoop and Gush lower entry barrier for distributed systems innovation
  • Undergrads are capable of doing great work!
Thanks!

• More info:
  • http://www.cs.williams.edu/~jeannie/cs339
  • jeannie@cs.williams.edu

• Gush
  • http://gush.cs.williams.edu