Bringing BIG Systems to small Schools

Jeannie Albrecht

Williams College

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

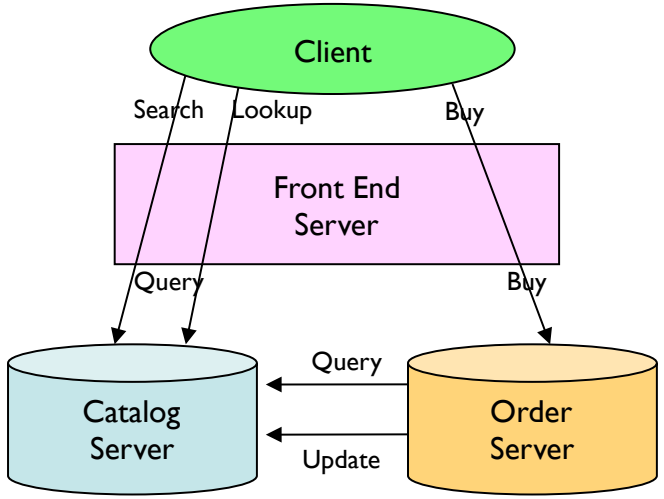
- Projects are 45% of overall grade
- Students work alone or with a partner
- Projects designed to emphasize techniques and technology from lecture topics and reading assignments
- Projects include a technical writing component
- Explored four different architectural models: client-server, multi-tier 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
- Goals
  - Explore simple client-server distributed computing paradigm
  - Gain experience with network/socket programming
  - Compare and contrast performance of HTTP1.0 and HTTP1.1 under varying conditions
- Student performance
  - All finished within 2.5 weeks; all projects worked
  - Quality of code varied significantly (some students had little prior experience with C)
  - Some write-ups were poorly written

Project 2: Online Bookstore

- Assignment: Build a multi-tier online bookstore
  - Use Java and Python
  - Use ~~Java RMI~~ XML-RPC
  - Ensure proper synchronization
- Goals
  - Explore multi-tier distributed computing paradigm
  - Gain experience with RPCs and network programming in Java
  - Evaluate performance under varying levels of load
- Student performance
  - All easily finished within 2 weeks
  - Quality of code was good overall
  - Write-ups were slightly better



Project 3 v1: Inverted Index

- Assignment: Build an inverted index using Hadoop
  - (Hadoop is open-source implementation of Google's MapReduce framework for large-scale data processing)
  - Return valid mapping of words to documents using eBooks from Project Gutenberg as input
- Setup
  - Created 60+ Xen virtual machines to host Hadoop mini-clusters using 14 cluster machines
  - Students maintained/configured their own cluster
- Goals
  - Explore "cutting-edge" cluster computing paradigm
  - Gain experience with basic system administration
- Student performance
  - All finished within 3 weeks
  - Good write-ups

Project 3 v1.5: Inverted Index

- Assignment: Build an inverted index using Hadoop
  - (Hadoop is open-source implementation of Google's MapReduce framework for large-scale data processing)
  - Return valid mapping of words to documents using eBooks from Project Gutenberg as input
- Setup
  - Awarded Amazon teaching grant
  - Created clusters on Amazon EC2 platform
  - Students maintained/configured their own cluster
- Goals
  - Explore "cutting-edge" cluster computing paradigm
  - Gain experience with basic system administration
- Student performance
  - All finished within 3 weeks
  - Good write-ups

Project 3 v2: Contextual Advertising

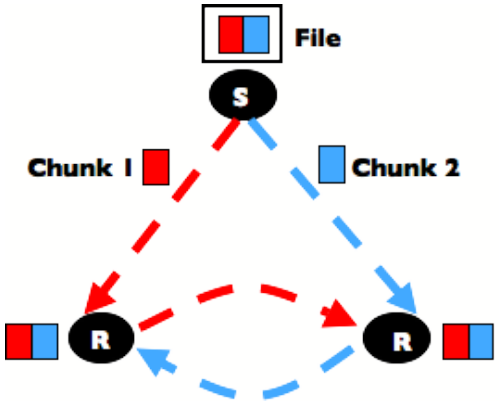
- Assignment: Given an advertising context, predict which advertisement is most likely to be clicked
  - Designed by another student (my TA) during W12
  - Compute click-through rate for ad id and page URL
- Setup
  - Awarded Amazon teaching grant
  - Created small clusters on Amazon EC2 platform
  - Dataset also comes from Amazon
  - Students maintained/configured their own cluster
- Goals
  - Explore "cutting-edge" cluster computing paradigm
  - Gain experience with basic system administration
- Student performance
  - All finished within 3 weeks
  - Good write-ups

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
- Setup
  - Created each group their own PlanetLab slice
  - Showed students how to use Plush for app management
- Goals
  - Explore P2P wide-area distributed computing paradigm
  - Allow students freedom to innovate
- Student performance
  - All finished and presented work within 3.5 weeks
  - Number of machines used ranged from 12 to 400+
  - Excellent write-ups

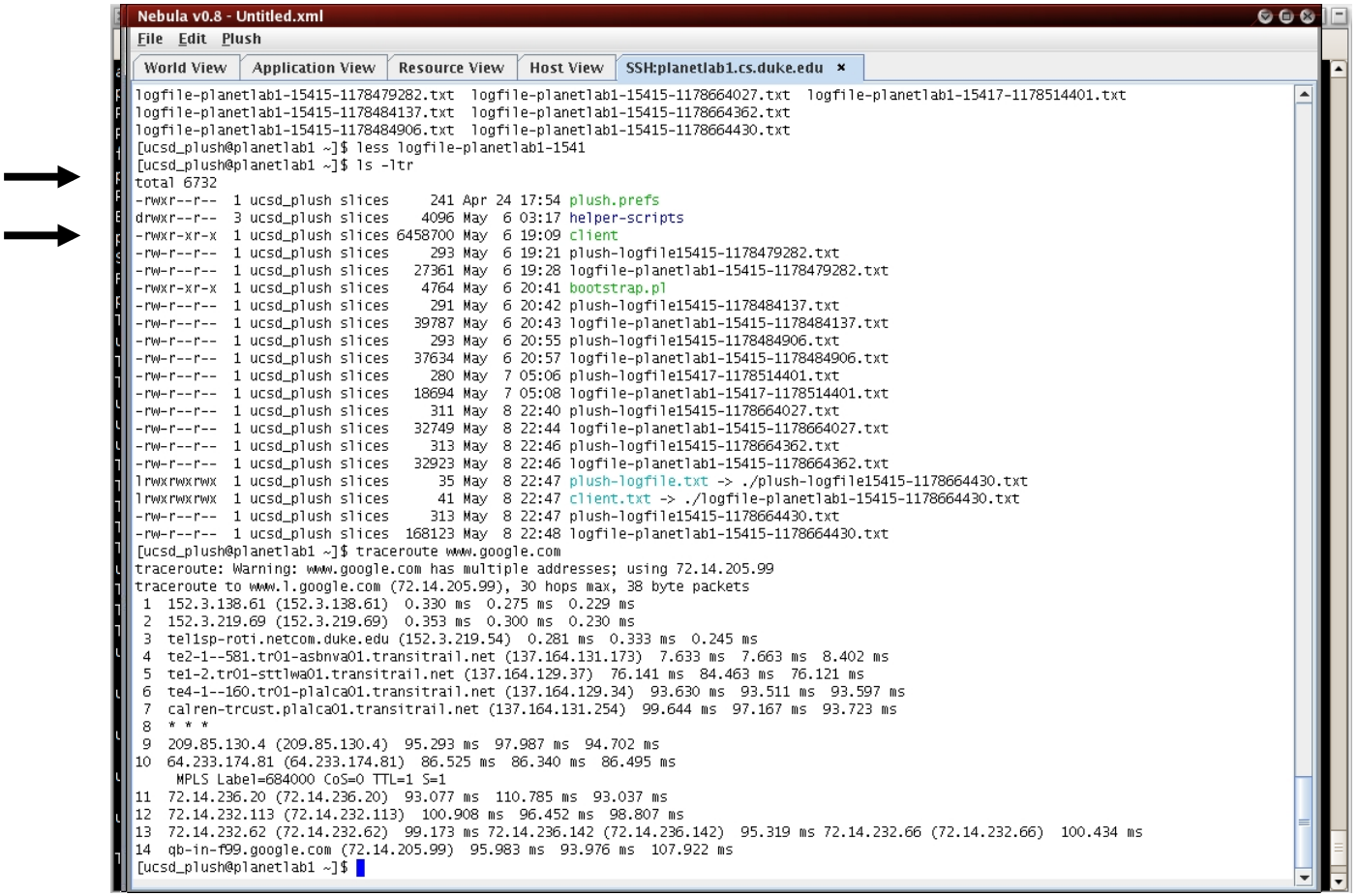
Project 4 v2: Final Project

- Assignment: Open-ended final project
- "Default" project: Build a P2P file-sharing system
  - Run system on PlanetLab
  - Setup
    - Created each group their own PlanetLab slice
    - Many students used Plush/Gush for app management
  - Goals
    - Explore P2P wide-area distributed computing paradigm
    - Allow students freedom to innovate
- Student performance
  - All finished and presented work within 4 weeks
  - Excellent write-ups



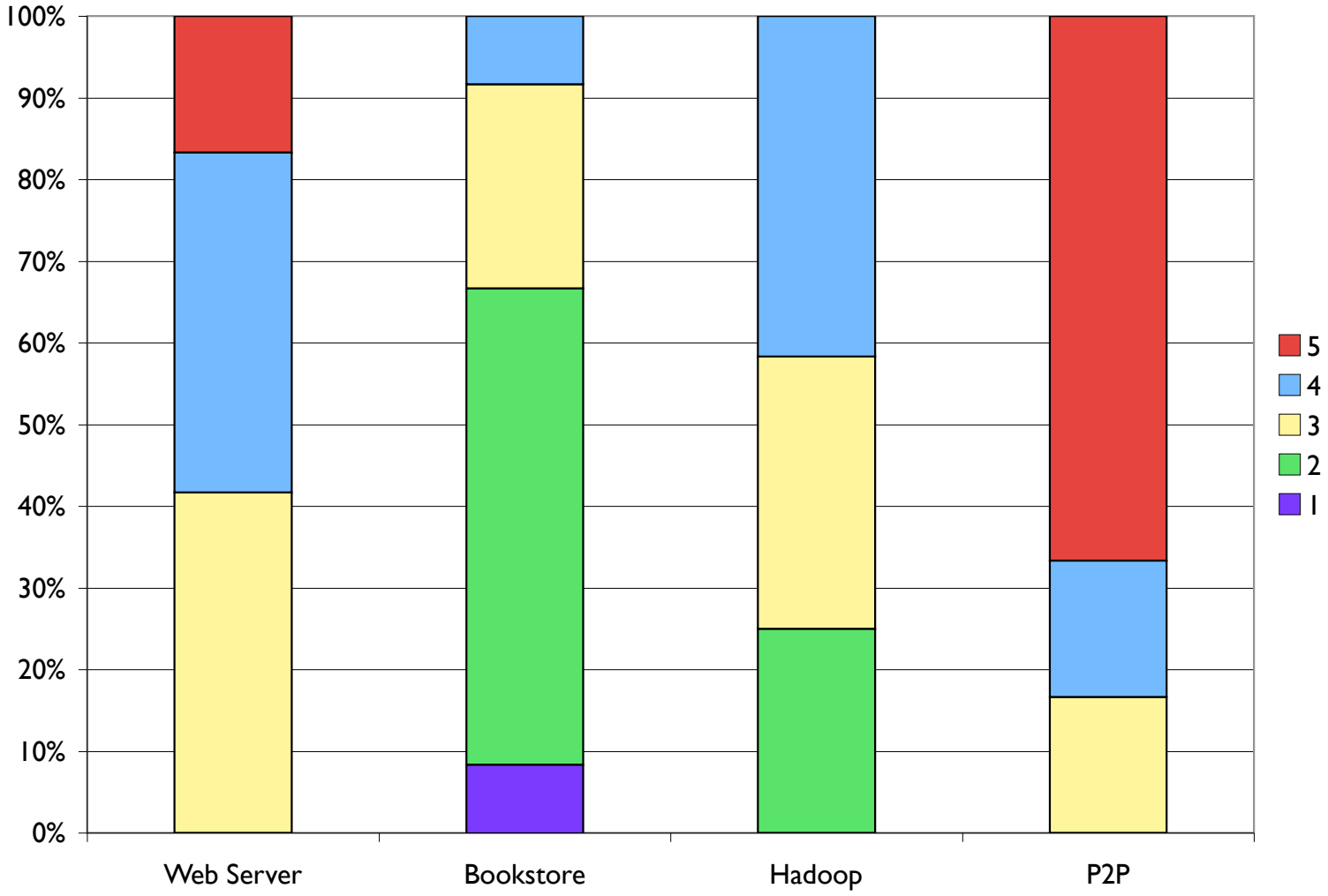
Plush/Gush User Interfaces

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

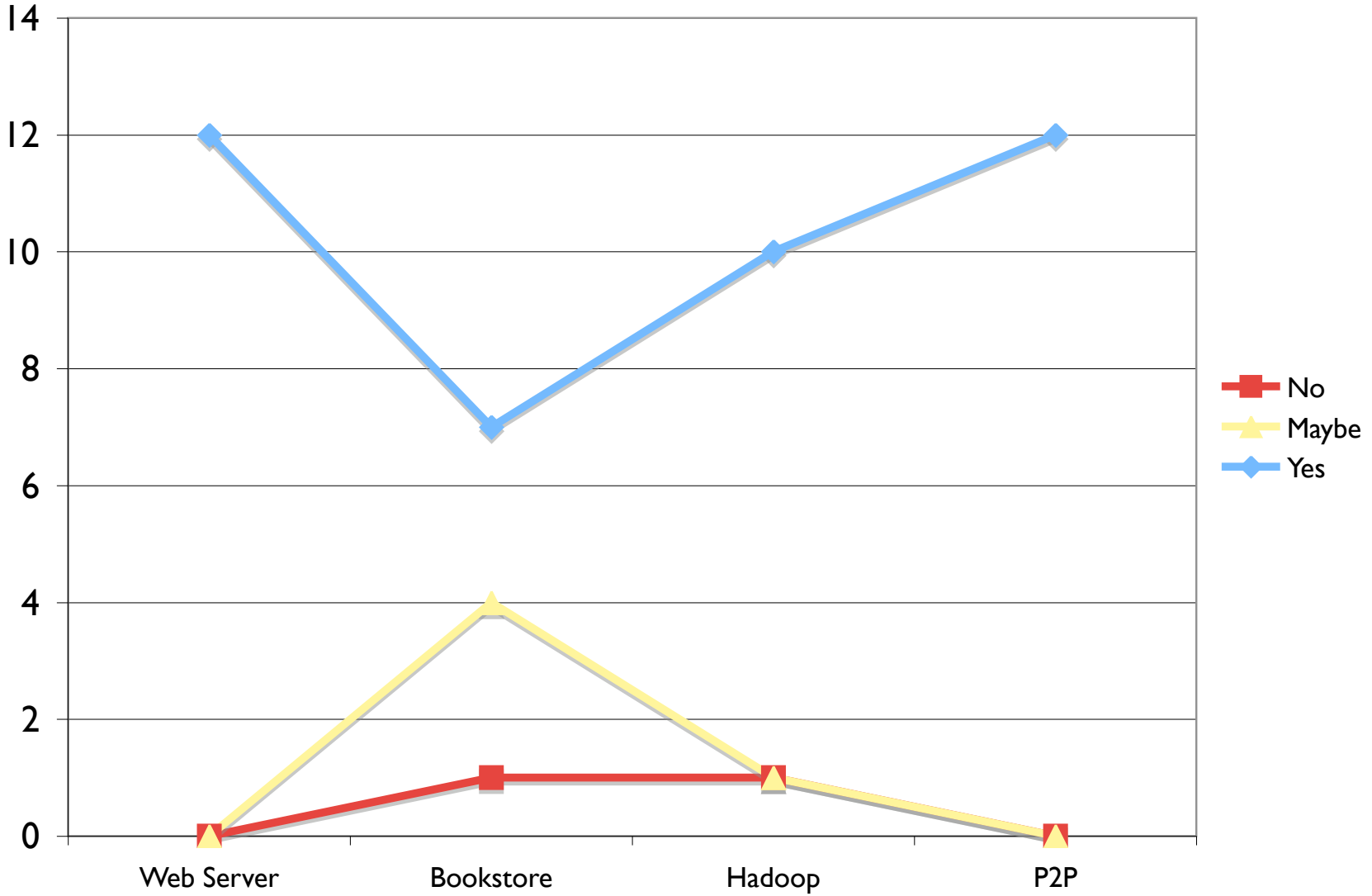


Project Difficulty (S08)

- 5 = difficult, 1 = easy



Recommend Again? (S08)



Student Feedback

- "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."
- "[The P2P 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."

Instructor Feedback

- Students loved Projects 1 and 4 (one student turned Project 4 into a senior thesis)
  - Projects 2 and 3 were too easy initially; better now
  - Some students loved open-endedness of Project 4; some struggled with it (default project helps)
  - Need about 4 weeks for final project
- I spent 4-5 hours per week in lab helping students
  - Students worked an avg of 10 hours per week
- Good writers != good technical writers
  - Students need practice writing technical/scientific papers!
- Students enjoy reading research papers
  - …when they understand the content

Conclusions

- We should teach undergraduates how to design and implement Distributed Systems!
- Shared computing platforms provide students with the opportunity to gain hands-on experience with large-scale, wide-area distributed computing
  - Use shared platforms as learning laboratories
  - Bring tech-richness of big universities to small colleges
- Frameworks like Hadoop, Plush/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
- PlanetLab / GENI
  - http://www.planet-lab.org
  - http://www.geni.net
- Plush and Gush
  - http://plush.cs.williams.edu
  - http://gush.cs.williams.edu