

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

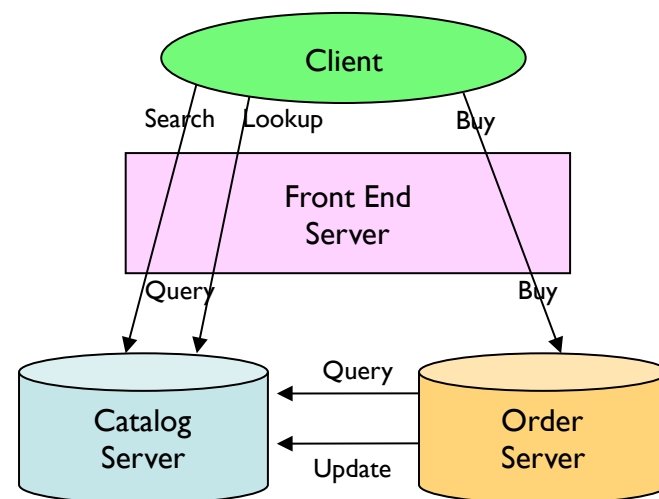
- Projects are 45% of overall grade
- Students work alone or with a partner
- Projects designed to emphasize techniques and technology from lecture topics and reading assignments
- Projects include a technical writing component
- Explored four different architectural models: client-server, multi-tier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

Project I: 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



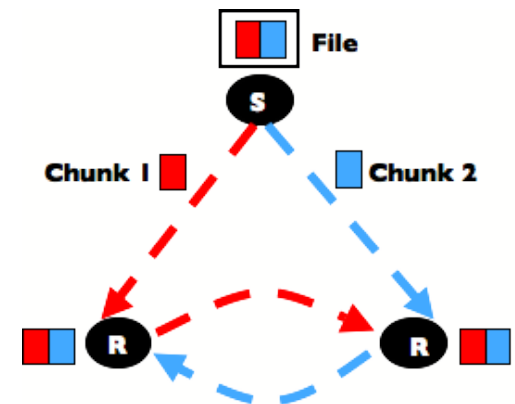
Project 4 v1: P2P Computing

- 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

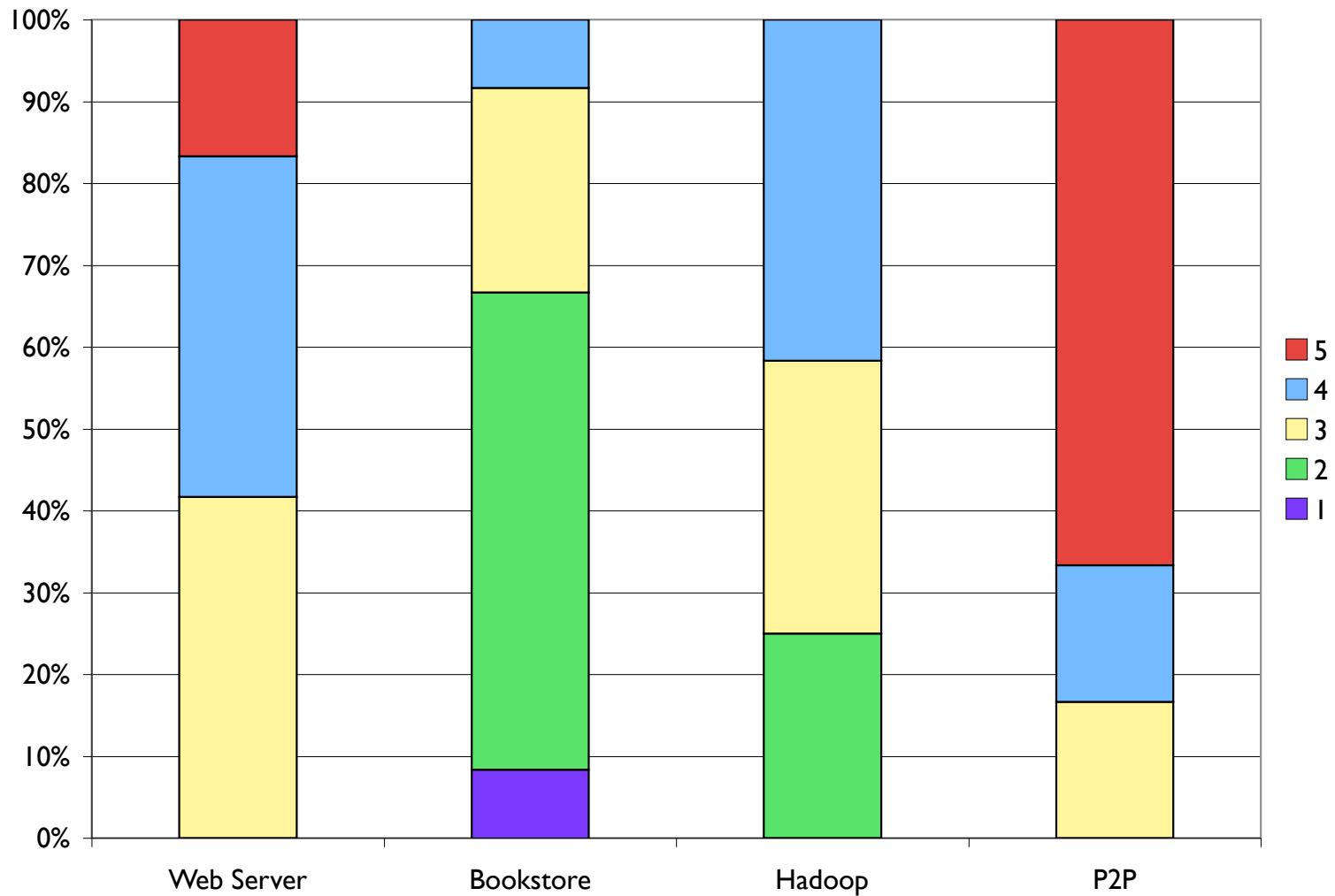
- Command-line interface used to interact with applications
- Nebula (GUI) allows users to describe, run, & visualize applications
- XML-RPC interface for managing applications programmatically



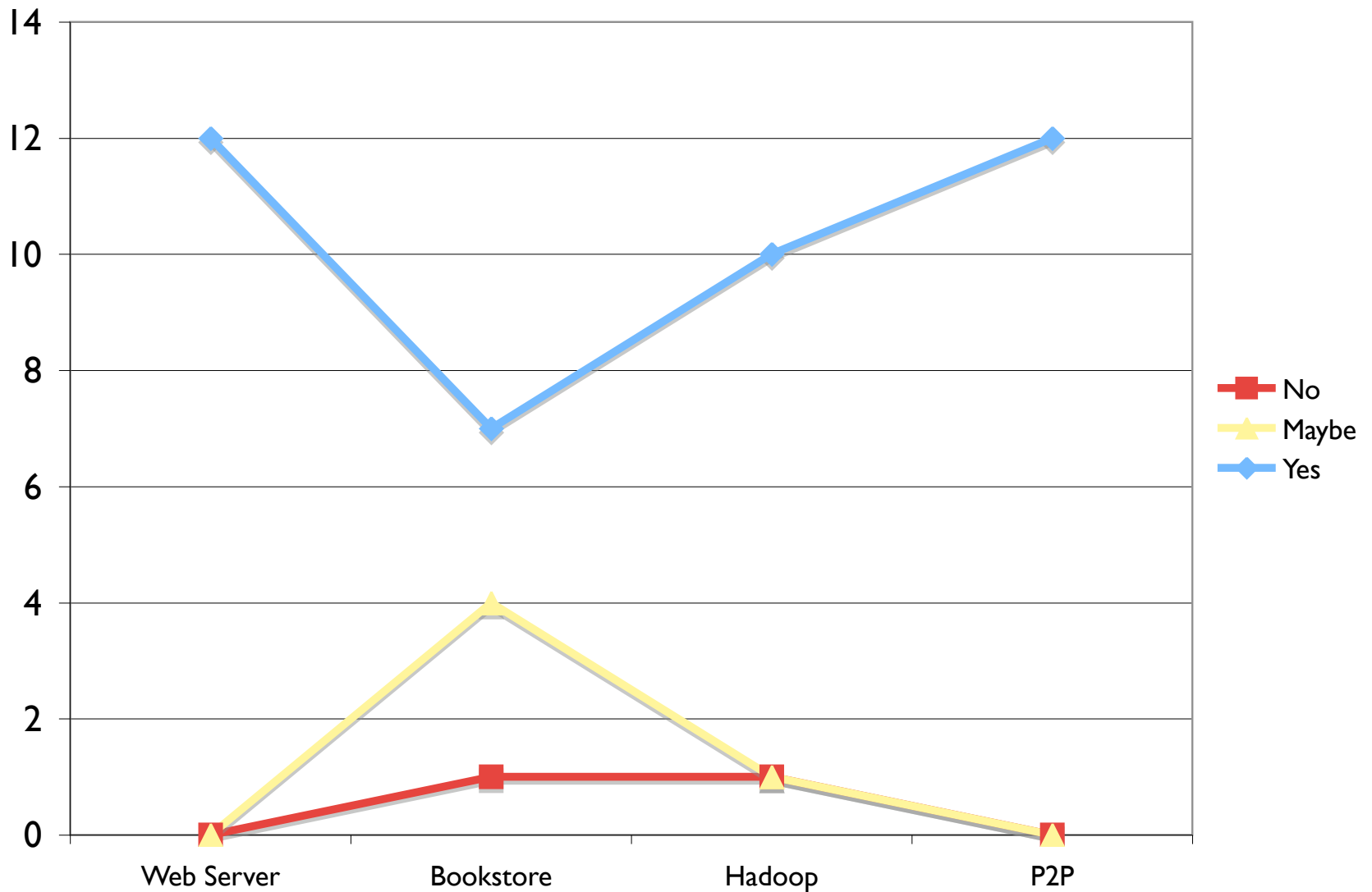
```
Nebula v0.8 - Untitled.xml
File Edit Plush
World View Application View Resource View Host View SSH:planetlab1.cs.duke.edu *
logfile-planetlab1-15415-1178479282.txt logfile-planetlab1-15415-1178664027.txt logfile-planetlab1-15417-1178514401.txt
logfile-planetlab1-15415-1178484137.txt logfile-planetlab1-15415-1178664362.txt
logfile-planetlab1-15415-1178484906.txt logfile-planetlab1-15415-1178664430.txt
[ucsd_plush@planetlab1 ~]$ less logfile-planetlab1-1541
[ucsd_plush@planetlab1 ~]$ ls -ltr
total 6732
-rwxr--r-- 1 ucsd_plush slices 241 Apr 24 17:54 plush.prefs
drwxr--r-- 3 ucsd_plush slices 4096 May 6 03:17 helper-scripts
-rwxr-xr-x 1 ucsd_plush slices 6458700 May 6 19:09 client
-rw-r--r-- 1 ucsd_plush slices 293 May 6 19:21 plush-logfile15415-1178479282.txt
-rw-r--r-- 1 ucsd_plush slices 27361 May 6 19:28 logfile-planetlab1-15415-1178479282.txt
-rwxr-xr-x 1 ucsd_plush slices 4764 May 6 20:41 bootstrap.pl
-rw-r--r-- 1 ucsd_plush slices 291 May 6 20:42 plush-logfile15415-1178484137.txt
-rw-r--r-- 1 ucsd_plush slices 39787 May 6 20:43 logfile-planetlab1-15415-1178484137.txt
-rw-r--r-- 1 ucsd_plush slices 293 May 6 20:55 plush-logfile15415-1178484906.txt
-rw-r--r-- 1 ucsd_plush slices 37634 May 6 20:57 logfile-planetlab1-15415-1178484906.txt
-rw-r--r-- 1 ucsd_plush slices 280 May 7 05:06 plush-logfile15417-1178514401.txt
-rw-r--r-- 1 ucsd_plush slices 18694 May 7 05:08 logfile-planetlab1-15417-1178514401.txt
-rw-r--r-- 1 ucsd_plush slices 311 May 8 22:40 plush-logfile15415-1178664027.txt
-rw-r--r-- 1 ucsd_plush slices 32749 May 8 22:44 logfile-planetlab1-15415-1178664027.txt
-rw-r--r-- 1 ucsd_plush slices 313 May 8 22:46 plush-logfile15415-1178664362.txt
-rw-r--r-- 1 ucsd_plush slices 32923 May 8 22:46 logfile-planetlab1-15415-1178664362.txt
lrwxrwxrwx 1 ucsd_plush slices 35 May 8 22:47 plush-logfile.txt -> ./plush-logfile15415-1178664430.txt
lrwxrwxrwx 1 ucsd_plush slices 41 May 8 22:47 client.txt -> ./logfile-planetlab1-15415-1178664430.txt
-rw-r--r-- 1 ucsd_plush slices 313 May 8 22:47 plush-logfile15415-1178664430.txt
-rw-r--r-- 1 ucsd_plush slices 168123 May 8 22:48 logfile-planetlab1-15415-1178664430.txt
[ucsd_plush@planetlab1 ~]$ traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 72.14.205.99
traceroute to www.l.google.com (72.14.205.99), 30 hops max, 38 byte packets
 1 152.3.138.61 (152.3.138.61) 0.330 ms 0.275 ms 0.229 ms
 2 152.3.219.69 (152.3.219.69) 0.353 ms 0.300 ms 0.230 ms
 3 telisp-roti.netcom.duke.edu (152.3.219.54) 0.281 ms 0.333 ms 0.245 ms
 4 te2-1--581.tr01-asbnva01.transitrail.net (137.164.131.173) 7.633 ms 7.663 ms 8.402 ms
 5 te1-2.tr01-sttlwa01.transitrail.net (137.164.129.37) 76.141 ms 84.463 ms 76.121 ms
 6 te4-1--160.tr01-plalca01.transitrail.net (137.164.129.34) 93.630 ms 93.511 ms 93.597 ms
 7 calren-trcust.plalca01.transitrail.net (137.164.131.254) 99.644 ms 97.167 ms 93.723 ms
 8 * * *
 9 209.85.130.4 (209.85.130.4) 95.293 ms 97.987 ms 94.702 ms
10 64.233.174.81 (64.233.174.81) 86.525 ms 86.340 ms 86.495 ms
   MPLS Label=684000 CoS=0 TTL=1 S=1
11 72.14.236.20 (72.14.236.20) 93.077 ms 110.785 ms 93.037 ms
12 72.14.232.113 (72.14.232.113) 100.908 ms 96.452 ms 98.807 ms
13 72.14.232.62 (72.14.232.62) 99.173 ms 72.14.236.142 (72.14.236.142) 95.319 ms 72.14.232.66 (72.14.232.66) 100.434 ms
14 qb-in-f99.google.com (72.14.205.99) 95.983 ms 93.976 ms 107.922 ms
[ucsd_plush@planetlab1 ~]$
```

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>