Distributed Data-Intensive Systems

CSCE 438
Texas A&M University
Department of Computer Science and Engineering
Instructor: Prof. James Caverlee

That’s all folks!
April 24, 2014
Distributed Objects Programming?
• Principles of distributed computing and programming with current paradigms, protocols, and application programming interfaces including Sockets, RMI, CORBA, IDL, Servlets, Web Services; security issues with public/private keys, digital signatures, forms and GUI based applications with multi-tier components, database connectivity and storing/streaming data structured using XML.
This course

• Introduction to the principles of distributed computing and programming in the context of the emerging data-centric view of computing (popularized by recent discussions of Big Data and Cloud Computing).
Four Big Course Themes

• Peer-to-peer systems
• Web-based systems
• Cloud computing
• Crowd-powered systems (crowdsourcing)
Distributed Data-Intensive Systems
=
People
P2P Systems

• P2P: Background and Brief Overview

• The Beginning
  • Napster

• Major Thrust 1: Unstructured P2P Networks
  • Gnutella
  • FastTrack/Kazaa (Foundations: Super-peers)

• Major Thrust 2: Structured P2P Networks
  • Foundations: Consistent Hashing
  • CHORD

• Today’s Champion:
  • BitTorrent (Foundations: Game theory; tit-for-tat)

• Open Issues:
  • EigenTrust
Shawn Fanning
Beverly Yang
Crowd-Powered Systems

- Crowd: Origins and Examples
- Wisdom of the Crowd
  - When it works + Failures of Crowds
- Crowdsourcing
  - Amazon Mechanical Turk
  - Case studies
  - Incentives (Red Balloon, money, …)
  - Quality control
- Mining Collective Intelligence
  - Google Flu trends
  - Classification: Decision Trees
- Prediction Markets
- The Future of Crowd Work
Cloud Computing

- The Cloud
  - Introduction and Overview
- Practicalities
  - Amazon Web Services
- Key Data Structure: Bloom Filters
- MapReduce
- NoSQL / Cloud storage
What’s next …

- Life-long learning

- (seriously)
- Stay in tune with current developments, e.g.:
  - http://www.humancomputation.com/2013/
  - http://www.allthingsdistributed.com
  - http://www.pytexas.org/2014/
  - pick a paper we read and follow the authors
  - + other resources (particularly ones posted on Piazza)

- Do research!
Preparing for the final

- Core technical stuff (75 to 80%)
  - Querying Napster vs. Gnutella vs. Super-peer network
  - Consistent hashing and CHORD rings
  - Tit-for-tat (and other variations)
  - Red balloon incentive scheme
  - Decision trees + Gini coefficient
  - Bloom filters
Preparing for the final

- History of failures (15% ish)
  - Example: What is Napster? Why did it fail?
  - Example: What is Google Flu? Why did it fail?
  - Example: What is crowdsourcing? And why did it fail in the Boston bomber incident?

- More seriously:
  - What are factors contributing to quality AMT work?
  - ...
Preparing for the final

• Maybe one or two “design” questions (10% ish)
Preparing for the final

• What is NOT covered?
  • MapReduce
  • Cloud Storage