Distributed Systems

Day 1
Course Staff!

- **Instructor**
  - Theophilus Benson (tab@cs.brown.edu)

- **HTAs**
  - Joshua Pettiz (jpattiz@cs.brown.edu)
  - Martin Ma (zma17@cs.brown.edu)

- **TAs**
  - Ali Mir (am209@cs.brown.edu)
  - Brian Oppenheim (boppenhe@cs.brown.edu)
  - Galadriel Brady (gbrady1@cs.brown.edu)
  - Kerem Gurbey (kerem_gurbey@brown.edu)
  - Kristen McLean (kmclean1@cs.brown.edu)
  - Tina Lu (yuyang_lu@brown.edu)
  - William Riley (wriley1@cs.brown.edu)
  - Zhedi Zhang (zzhang57@cs.brown.edu)
About Me!

• Theophilus Benson
  • Office: CIT 327
  • Office Hours: Tues: 4-5pm, Wed:11-12am

• Research Interests:
  • Networking for Developing Regions
  • ML for Systems
  • Highly Available Infrastructures

• Non-Research Interests:
  • Climbing ..... (being horrible at Climbing)

• Academic
  • PhD U of Wisconsin (2012)
  • PostDoc Princeton (2013)
  • Prof @ Duke (2013-2017)

• Industry
  • Microsoft (2008-2010)
  • Facebook (2014 & Now)
A *distributed system* is a system whose *components* are located on different networked *computers* ... The components interact with one another in order to achieve a common goal.
Why Do You Need a Distributed System?
IoT (Google Version)

https://cloud.google.com/iot-core/images/benefits-diagram.png
TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. It maps the nodes of a dataflow graph across many machines in a cluster, and within a machine across multiple computational devices...

TensorFlow Architecture

https://www.tensorflow.org/guide/extend/architecture
BitCoin: BFT

Byzantine Fault Tolerance: The Key for Blockchains

June 29, 2017, 12:03:44 PM EDT By Christopher Tozzi, Distributed

https://www.nasdaq.com/article/byzantine-fault-tolerance-the-key-for-blockchains-cm810058
Google Case Study

http://www.google.com
Google Case Study

• Goal: Rank webpages
  • Index and store large data
  • @2009: 4+ PB Filesystems

• Problem with one server
  • Can’t store all the data
  • Can’t process all the data

• @2013, 12 DCs around the world

http://www.google.com
Facebook Case Study

https://www.facebook.com/notes/facebook-engineering/scaling-out/23844338919
Facebook Case Study

• Goal: Quickly show ads to users
  • Users are spread out around the world

• Problem: traffic can’t go faster than the speed of light

• @2007 FB, Opens a second DC

https://www.facebook.com/notes/facebook-engineering/scaling-out/23844338919
Benefits of a Distributed System

• Performance

• Scalability

• Resource Sharing

• Fault Tolerance
This Class

Algorithms and designs to enables a set of components to behave as one during arbitrary failures modes

Challenges:
- Availability
- Performance
- Consistency
- Security
Course Information
Distributed Systems Are Hard to Build!

Challenges:
- Availability
- Performance
- Consistency
- Coordination
- Security

Algorithms and designs to enables a set of components to behave as one during arbitrary failures modes

Node Failures

Network Failures
Communication
(Naming, RPC, TCP/UDP)

Load Balancing: DHT
(Sharding, Replication, Partitioning)

Ordering: Time
(Global Snapshots)

Consensus: Agreement, Consistency, Coordination
(Active/Passive/Lazy Replication, Transactions....)

Distributed Systems in the Field
(Google, Amazon, Facebook, MongoDB)

1 week
1 week
1 week
3 weeks
4 weeks

Note: This accounts for 10 of the 12 weeks. The other 2 weeks include: reviews, help sessions, and labs sessions on gRPC and debugging
Communication
(Naming, RPC, TCP/UDP)

Load Balancing: DHT
(Sharding, Replication, Partitioning)

Ordering: Time
(Global Snapshots)

Consensus: Agreement, Consistency, Coordination
(Active/Passive/Lazy Replication, Transactions, ...)

Distributed Systems in the Field
(Google, Amazon, Facebook, MongoDB)
Textbook
Workload

• Homeworks (50%) (4 HWs)

• Projects (20%) (4 projects)

• Exams (30%)
  • Midterm (10%)
  • Final (20%)

• Class Partition (3% -- Bonus points)
  • Students @ Cusp Class Partition will help push over.

ReGrades

• Limit of 1 week (After grades returned)
  • Regrades may lead to a regrade
    • Points added!
    • Points deducted!
Skills Needed

• Ability to write and debug largish programs with processes
  • CS 32/33

• Ability to prove a theorem (or maybe two)
  • CS 22

• Willingness to learn a new programming language
  • You will be learning Go
Late Policy

• Each student has 4 free late days:
  • No more than 3 can be applied to any assignment
  • No more than 2 can be applied to the last assignment
  • For group projects, a late day (deducts from both students)

• Each day late == reduction of grade by 10% of total assignment points

• Exceptions: illness/Religious holiday
  • Illness: Provide note from health services/office of student life (Fill out Google form)
  • Religious holidays: Fill out Google form