Distributed Systems

Day 1: Introduction

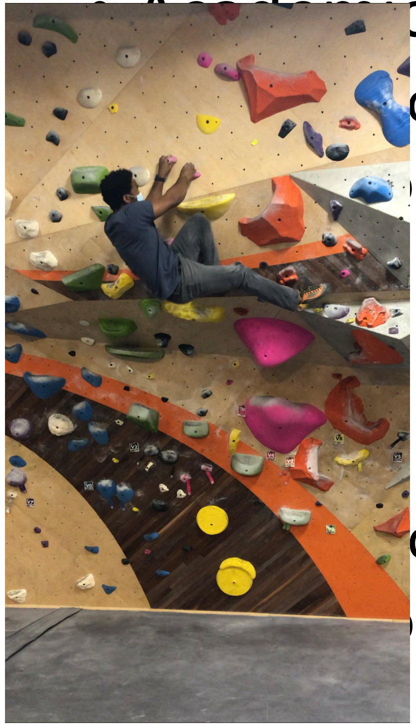
Jan 21, 2021

About Me!

- Theophilus Benson
  - Office: CIT 327 (on Zoom ☺)
  - Office Hours: 2:30-3:30 after class
- 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)

About Me!

- Theophilus Benson
  - Office: CIT 327 (on Zoom ☺)
  - Office Hours: 2:30-3:30 after class
- Research Interests:
  - Networking for Developing Regions
  - ML for Systems
  - Highly Available Infrastructures
- Non-Research Interests:
  - Climbing ….. (being horrible at Climbing)



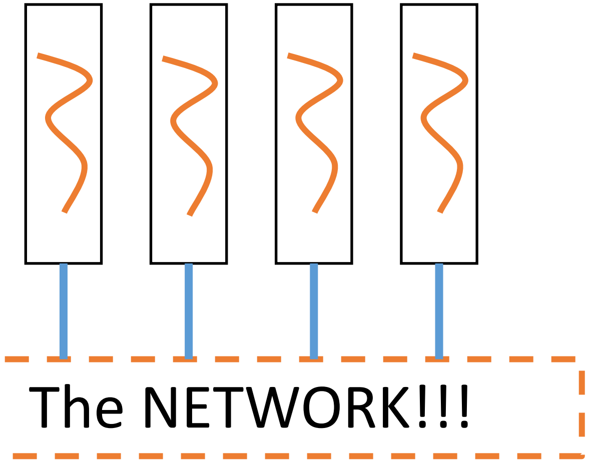
Academic
of Wisconsin (2012)
c F
Du
oft (2008-2010)
ok (2014 & Now)

Course Staff!

- Instructor
  - Theophilus Benson (tab@cs.brown.edu)
- HTAs
  - John Lhota
- TAs
  - Akshat Mahajan (amahaja5)
  - Johnny Roy (jroy8)
  - March Boonyapaluk (kboonyap)
  - Qingyi Li (qlu9)
  - Xiling Zhang (xzhan199)

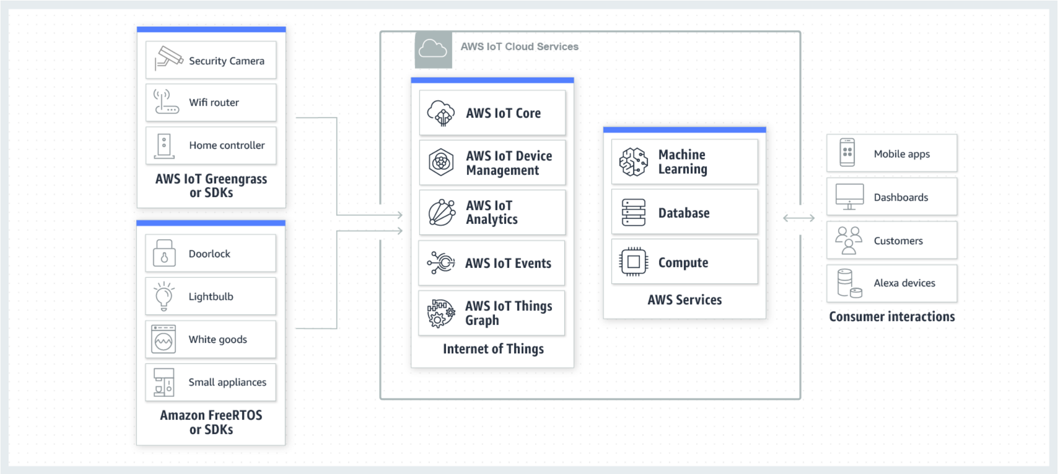
A ***distributed system*** is a system whose **components** are located on **different networked computers** ... The components interact with one another in order to achieve **a common goal**.

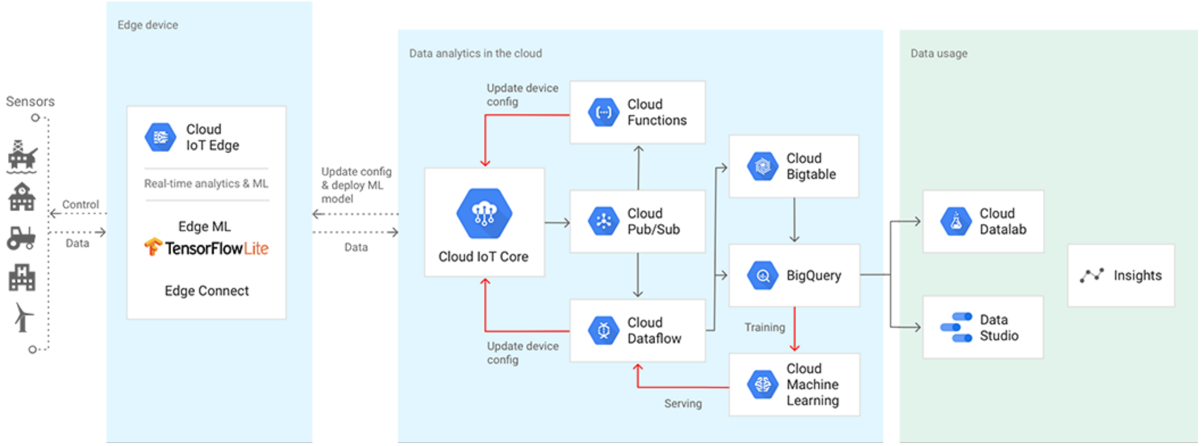
servers!!



Why Do You Need a Distributed System?

IoT Frameworks

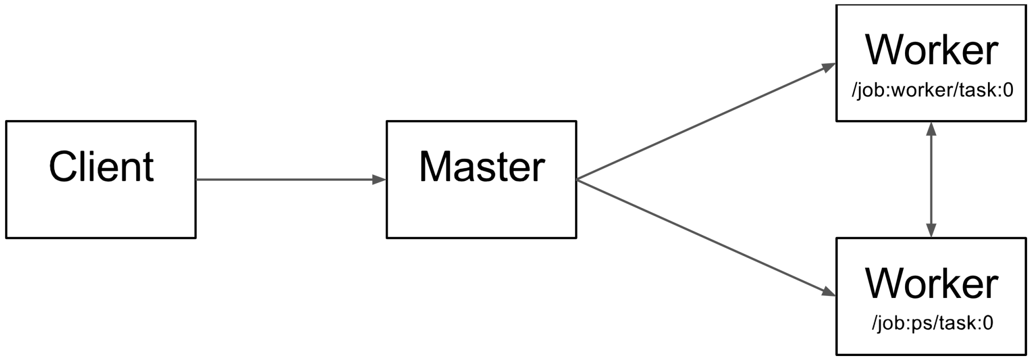




https://d1.awsstatic.com/r2018/b/IoT%20Category/AWS%20IoT%20Connected%20Home.933d5ce9339e86c9339fd833f5d50d3
https://cloud.google.com/iot-core/images/benefits-diagram.png

TensorFlow

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

TensorFlow: A system for large-scale machine learning

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng

Google Brain

Abstract

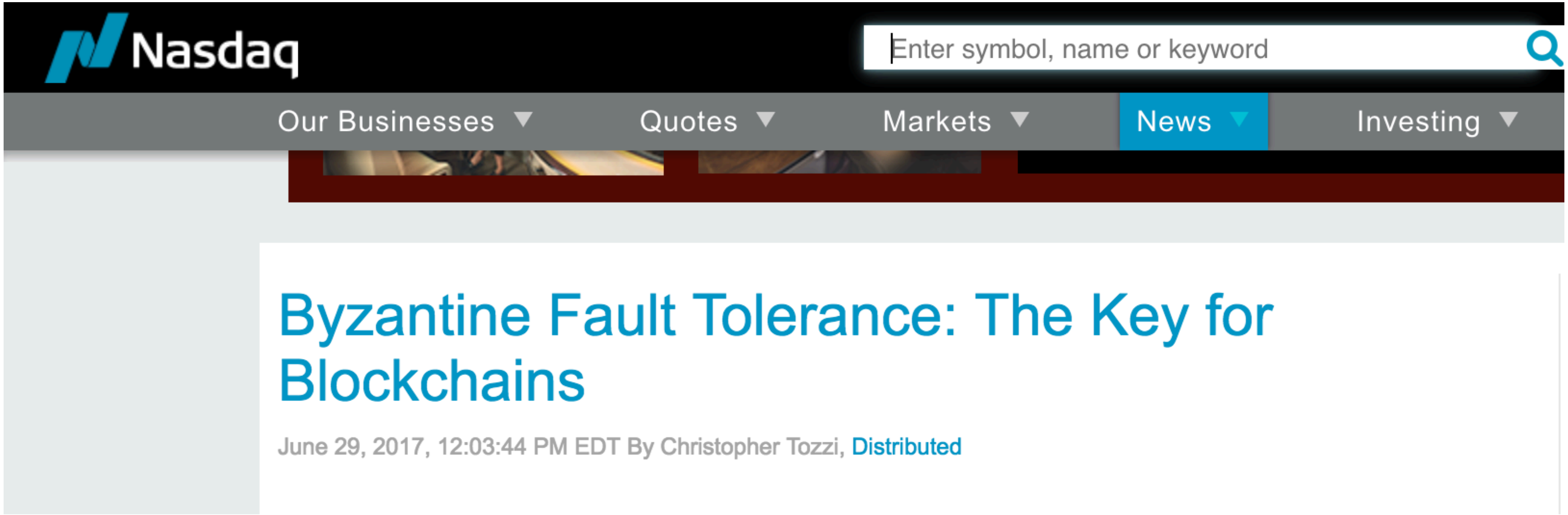
TensorFlow is a machine learning system that operates at large scale and in heterogeneous environments. TensorFlow uses dataflow graphs to represent computation, shared state, and the operations that mutate that state. It maps the nodes of a dataflow graph across many machines in a cluster, and within a machine across multiple computational devices, including multicore CPUs, general-purpose GPUs, and custom-designed ASICs known as Tensor Processing Units (TPUs). This architecture gives flexibility to the application developer: whereas in previous “parameter server” designs the management of shared state is built into the system, TensorFlow enables developers to experiment with novel optimizations and training algorithms. TensorFlow supports a variety of applications, with a focus on training and inference on deep neural networks. Several Google services use TensorFlow in production, we have released it as an open-source project, and it has become widely used for machine learning research. In this paper, we describe the TensorFlow dataflow model and demonstrate the compelling performance that Tensor-

datasets, and moving them into production. We have based TensorFlow on many years of experience with our first-generation system, DistBelief [20], both simplifying and generalizing it to enable researchers to explore a wider variety of ideas with relative ease. TensorFlow supports both large-scale training and inference: it efficiently uses hundreds of powerful (GPU-enabled) servers for fast training, and it runs trained models for inference in production on various platforms, ranging from large distributed clusters in a datacenter, down to running locally on mobile devices. At the same time, it is flexible enough to support experimentation and research into new machine learning models and system-level optimizations.

TensorFlow uses a unified dataflow graph to represent both the computation in an algorithm *and* the state on which the algorithm operates. We draw inspiration from the high-level programming models of dataflow systems [2, 21, 34] and the low-level efficiency of *parameter servers* [14, 20, 49]. Unlike traditional dataflow systems, in which graph vertices represent functional computation on immutable data, TensorFlow allows vertices to represent computations that own or update mutable state

https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf
https://www.tensorflow.org/guide/extend/architecture

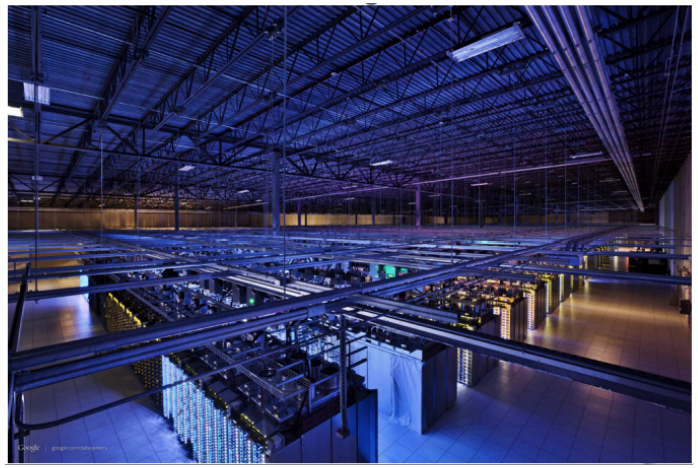
BitCoin: BFT



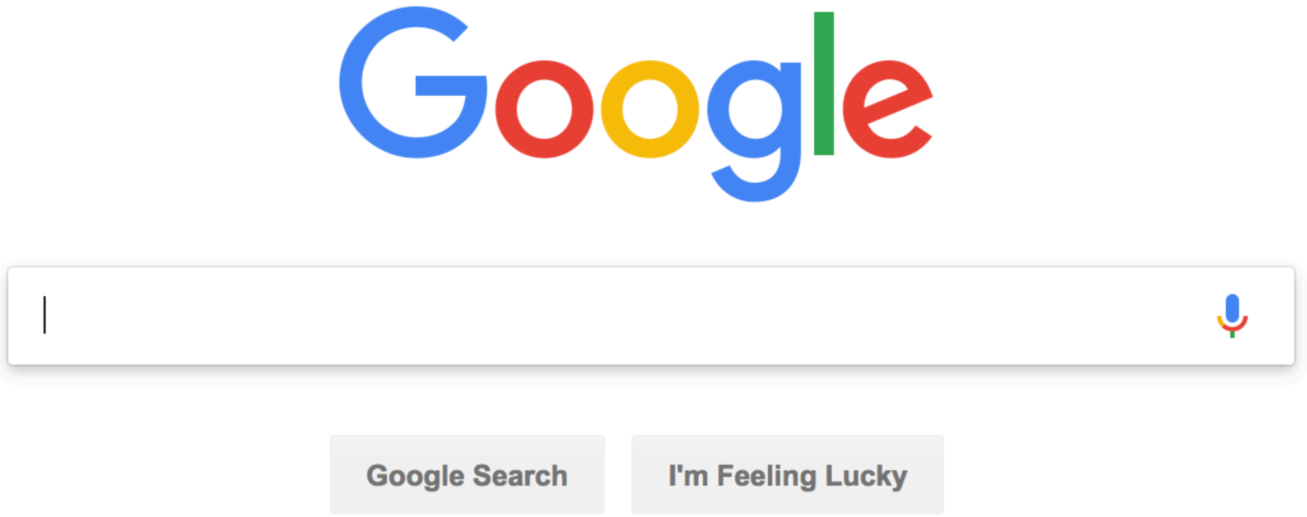
https://www.nasdaq.com/article/byzantine-fault-tolerance-the-key-for-blockchains-cm810058

Benefits of a Distributed System

- Performance
- Scalability
- Resource Sharing
- Fault Tolerance



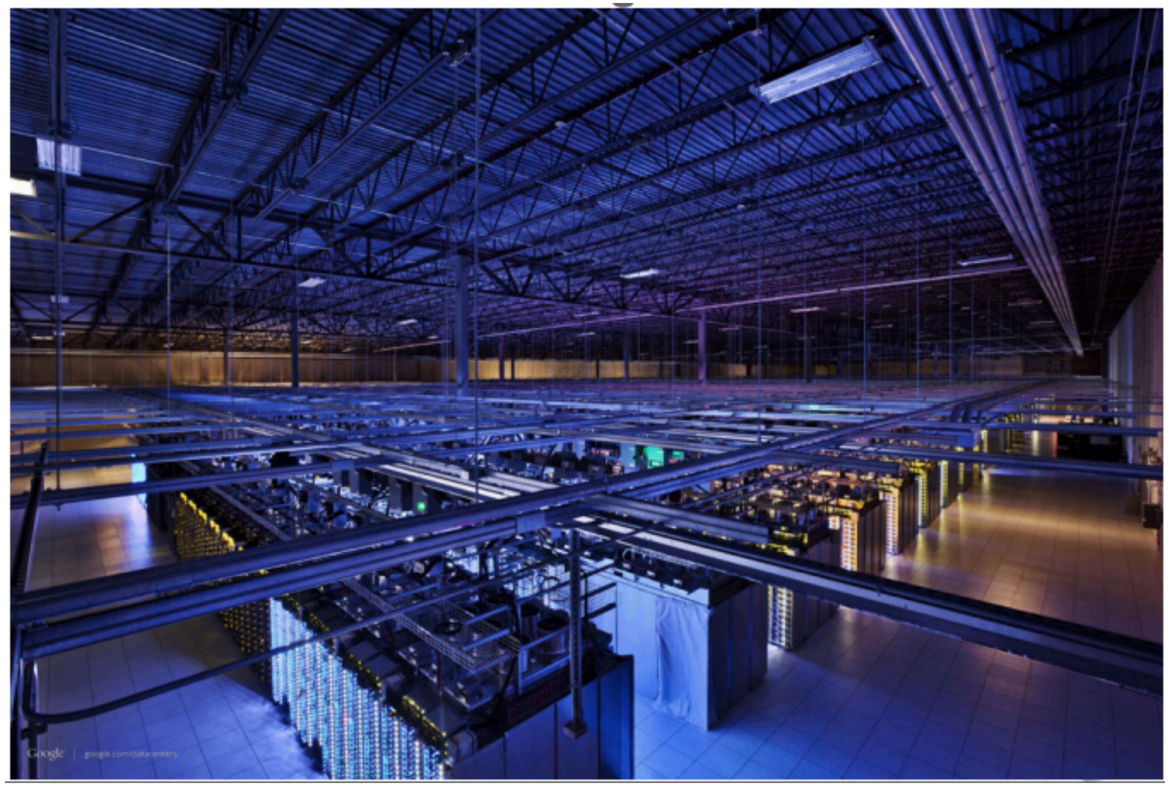
Google Case Study



http://www.google.com
http://infolab.stanford.edu/infoseminar/archive/WinterY2013/dean.pdf

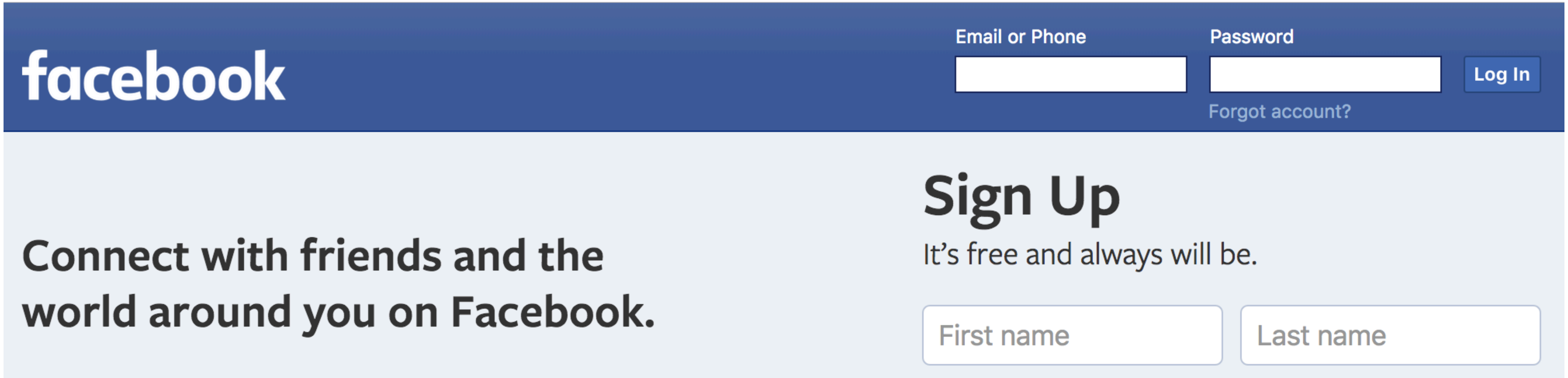
Google Case Study

- 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
http://infolab.stanford.edu/infoseminar/archive/WinterY2013/dean.pdf

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

This Class

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

Node Failures

Network Failures

Challenges:

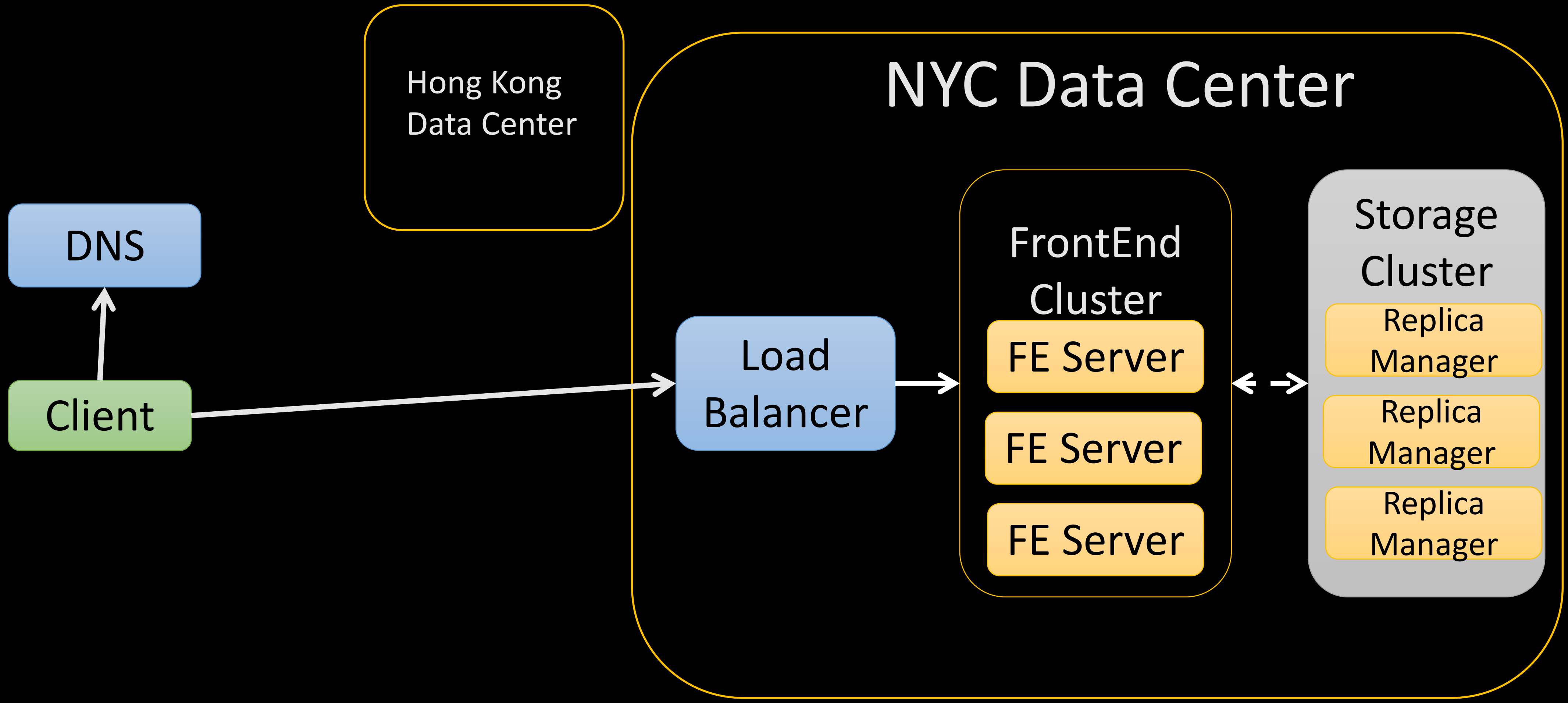
- Availability
- Performance
- Consistency
- Security

How do I design Google Drive?

What is a consistency model? What is the right consistency model?

How do I pick between performance and availability?

What is the appropriate mechanism for load balancing!



CS1380 – How to Design an Online Service Network

Course Information

Distributed Systems Are Hard to Build!

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

Node Failures

Network Failures

Challenges:

- Availability
- Performance
- Consistency
- Coordination
- Security

Communication
(Naming, RPC, TCP/UDP)

1 week

Load Balancing: DHT
(Sharding, Replication, Partitioning)

1 week

Ordering: Time
(Global Snapshots)

1 week

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

3 weeks

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

4 weeks

Note: This accounts for 10 of the 11 weeks.

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)

Proj 1: LiteMiner

Proj 2: Tapestry

Proj 3: Raft

Proj 4: Puddlestore

Each Project has labs to help prep you for the project

Workload

- Homeworks (15%) (6 HWs)
- Projects (50%) (4 projects)
- Exams (25%)
  - Midterm (10%)
  - Final (15%)
- Post lecture quizzes (7%)
- Class Partition (3%)
  - Students @ Cusp Class Partition will help push over.

Solo Assignments

Group Assignments

Closed Book

Re-Grades

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

Session 01: Synch

Session 02: ASynch

Session 03: Hybrid
MS Students only!
Meets Wed 12-12:50
CIT 477

Class Partition (3%)

TopHat/Zoom polls used during class .

For Asynch --- Canvas questions will be assigned.

Textbook


DISTRIBUTED SYSTEMS
Concepts and Design
Fifth Edition
George Coulouris
Jean Dollimore
Tim Kindberg
Gordon Blair

Capstone Requirements

- Reminder: Capstone is for seniors
- Capstone == Additional work on several on the projects!

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