Fundamentals of Large-Scale Distributed Systems
(a.k.a. Distributed Systems 1)

https://columbia.github.io/ds1-class/
Interested in...

1. scalable web services?
2. big data?
3. and the large-scale infrastructure systems making these possible?

If so, you're in the right room.
Distributed systems classes at CUCS

1. **Fundamentals of Distributed Systems** (DS1, RG, Fall)
   - 4000-level class
   - Teaches basic concepts, principles of large-scale, distributed-system design
   - Discusses application of those concepts/principles in real-world systems (e.g., Google/Facebook/Amazon)
   - Homework series teaches how to build basic DS

2. **Advanced Distributed Systems** (DS2, RG, Spring)
   - 6000-level class, i.e., research seminar
   - Discusses topics in depth by reading the most influential papers that originated them
   - We include both classical papers on well-understood topics and new, state-of-the-art papers
Related CU classes

• Multiple cloud computing/web programming/big data processing classes are offered @CU
  – Those classes teach you how to use various popular distributed systems
  – This class will teach you the how those and other systems are built, so you can build and use them better in the future

• Similar to the OS class, but for the distributed environment
  – And in the “cloud” era, everything is distributed!
    • If you want to do “big data,” you need DS
    • If you want to do mobile apps, you need DS
This class

1. **Foundational concepts** of large-scale distributed systems
   - Challenges, algorithms, techniques, abstractions

2. The **inner-workings** of several distributed systems serving as infrastructure for some **big companies**
   - E.g.: Google's protobuf/Spanner/MapReduce, Yahoo's Hadoop, Amazon's Dynamo, etc.
This class (cont)

3. **Gotchas** of using some popular distributed systems, which stem from their inner workings and reflect the **challenges** of building large-scale distributed systems
- MongoDB, Redis, Hadoop, etc.

4. **How to build** a real distributed system yourself!
   - Via a series of coding assignments, you will build your very own distributed system
1. **Distributed systems primer**: challenges and goals of distributed systems.
2. **Communication models**: remote procedure calls (RPC), RPC libraries, failure models, semantics.
3. **Time and coordination**: challenges, physical and logical clocks, distributed mutual exclusion.
4. **Sharding, replication, and the agreement problem**: commitment and consensus, use cases for each.
5. **Local transactions (background)**: ACID semantics, concurrency control, recovery mechanisms.
6. **Commitment protocols**: 2-phase commit, safety/liveness tradeoffs with the two.
7. **Consensus protocols**: Paxos overview, key ideas, basic algorithm, examples, liveness failure mode.
8. **Replication architectures**: fault-tolerant architectures (primary/secondary, master/slaves); the design of Google's Chubby lock service; the 2PC+Paxos approach to both scalability and fault tolerance.
9. **Case study: Google's Spanner**: design of TrueTime, design of Spanner and its fault-tolerant, linearizable, distributed transactions.
10. **Other consistency and isolation semantics**: sequential, causal, and eventual consistency; a few isolation semantics; mechanisms to achieve each; tradeoffs between them.
11. **Distributed computation**: MapReduce design, TensorFlow design, approximate computing engines.
12. **Distributed systems security primer**: authentication protocols, Needham-Schroeder, Kerberos, byzantine fault tolerance, maybe blockchain.
Important addresses

- Website: https://columbia.github.io/ds1-class/
- Discussions: Piazza and in class.
  - TAs will be active on Piazza, most homework questions should go there.
  - For questions on lecture materials, please come to class and pose them there!
Prerequisites

Pre-requisites:

- You must have a **solid** programming experience (C, C++, Java, preferably system-level programming experience)
- Columbia courses (or equivalents):
  - COMS W3137 Data Structures and Algorithms
  - COMS W3157 Advanced Programming
  - COMS W3827 Fundamentals of Computer Systems
  - Optional, but **very** useful: COMS 4118 Operating Systems

If you lack these prerequisites, do **not** take the class

- Heavy coding accounts for a large portion of the grade!
- Use Assignment 1 to figure out if you have sufficient experience
Grading

• Grading formula
  – 60%: four graded homeworks
  – 40%: midterm and final exams
  – 10%: extra credit for substantial contributions over an extended period of time in class or on Piazza

• Grading policies
  – You have 48 hours of lateness over the entire semester. Once you exceed those, submitting your homework one minute late will result in a zero for that late homework.
  – Can discuss, but *not* look at others' code or answers
  – Use Piazza to discuss
  – See policies on website, read them in detail!