CS219: Web and Mobile Systems

Lecture 1: Introduction

Ravi Netravali

http://web.cs.ucla.edu/~ravi/

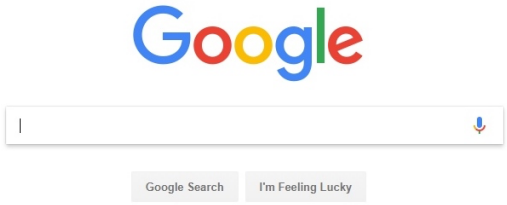
Today's Agenda

- Overview of topics

- Logistics
  - Class structure
  - Grading
  - Research project
  - Expectations and goals

Using Web/Mobile Apps

- We're constantly interacting with web/mobile systems, both actively…



Spotify

Skype

yelp

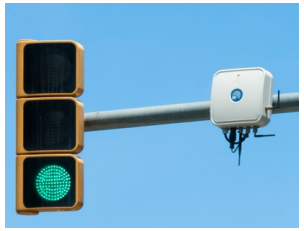
YouTube

- …and passively

fitbit

Google Maps

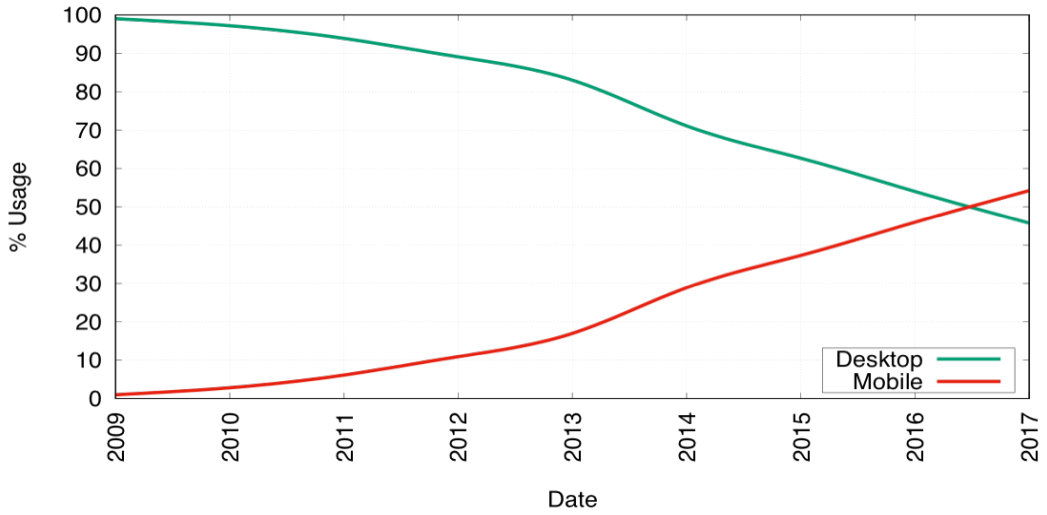
waze
OUTSMARTING TRAFFIC, TOGETHER



Web Trends

- We spend an average of 24 hours online each week
  - Up from 9.4 hours per week in 2000
- Our access mechanism has also changed: 84% of all internet access is from mobile devices

Web Usage



- Continual innovation: self-driving cars, smart homes

Relying on Apps

- We have also come to rely on these applications

Banking and Finance

E-commerce

News

Travel and navigation

Augmented reality

Each with different challenges and requirements:

- Security/privacy concerns
- Consistency
- Speed

All operate over:

- Client device/app
- Network
- Server app systems

Course Overview

- Network changes
  - Cellular, Wifi, BLE
- Device changes
  - Phones, tablets, smart devices
- Paradigm changes
  - Users create more, collaborative
- Richer services
  - Not just browsing: banking, entertainment, etc.

- Increases in:
  - Data
  - Performance demands
  - Sources of delay
  - Security vulnerabilities

How can we support this evolution securely and with high performance?

Course Overview

- Applications leverage active research in many areas!
  - Machine learning pipelines
  - New security/crypto techniques (including blockchains…)
  - Search algorithms
  - Debugging systems (e.g., provenance and data flow tracking)
  - Storage systems and memory models (e.g., NVM)
  - Network protocols (e.g., QUIC, BBR)

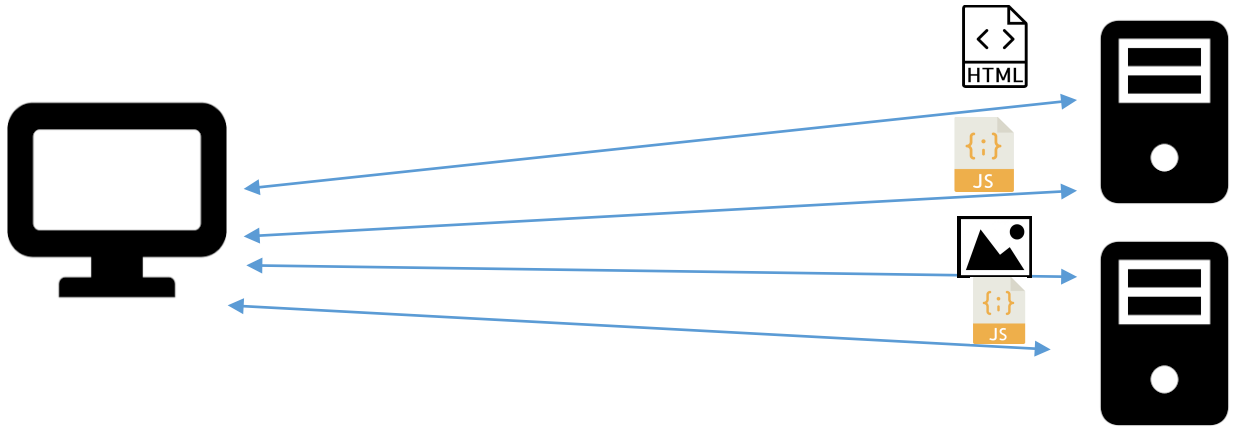
- How can we integrate these efficiently and securely?

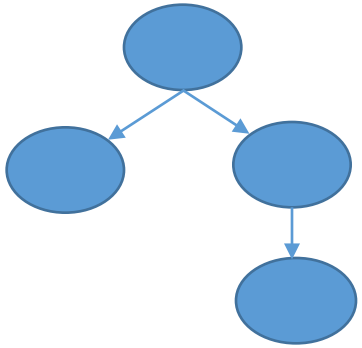
  Can significantly affect user experience (and revenue)!

Course Topics

Web Page Loads

Optimizations for faster page loads





Page Dependency Graph

Challenge: variable/constrained network, dependencies in loading page content

Solutions:

- Reorder requests to parallelize network overheads
- Send objects to clients ahead of time

Tradeoffs: security, overheads, infrastructure needs, etc.

Web Page Loads

Performance metrics: how to evaluate page loads?

Above The Fold

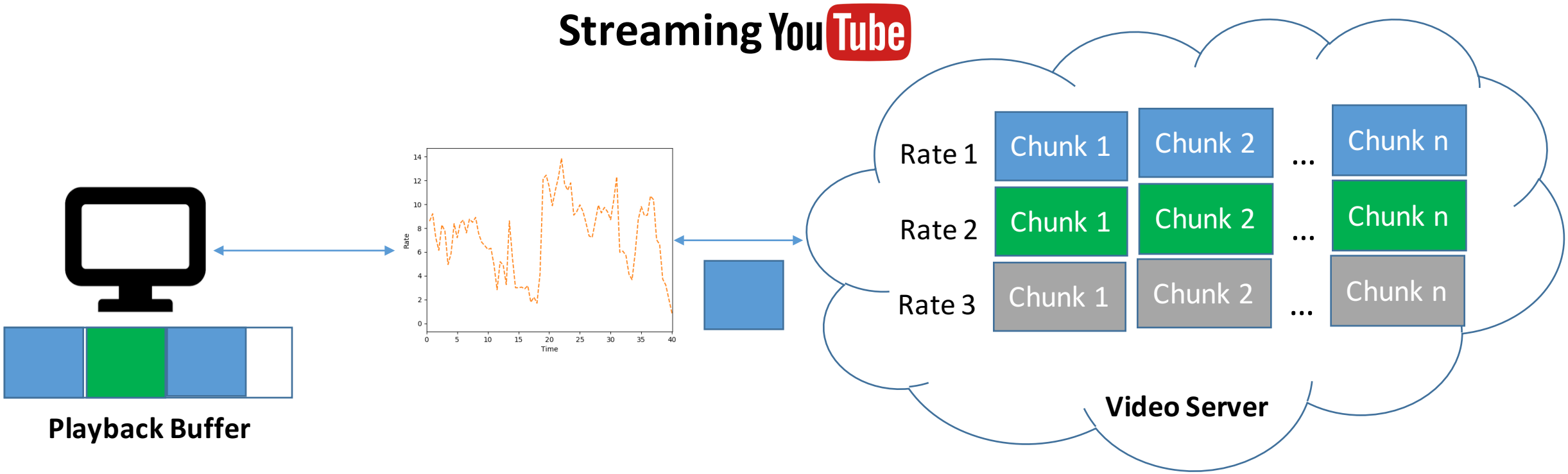
Below The Fold

Traditional page load time: everything loaded

Time-to-interactive: content for initial interaction loaded

Gaze: priority models based on user visual interactions with page

Video



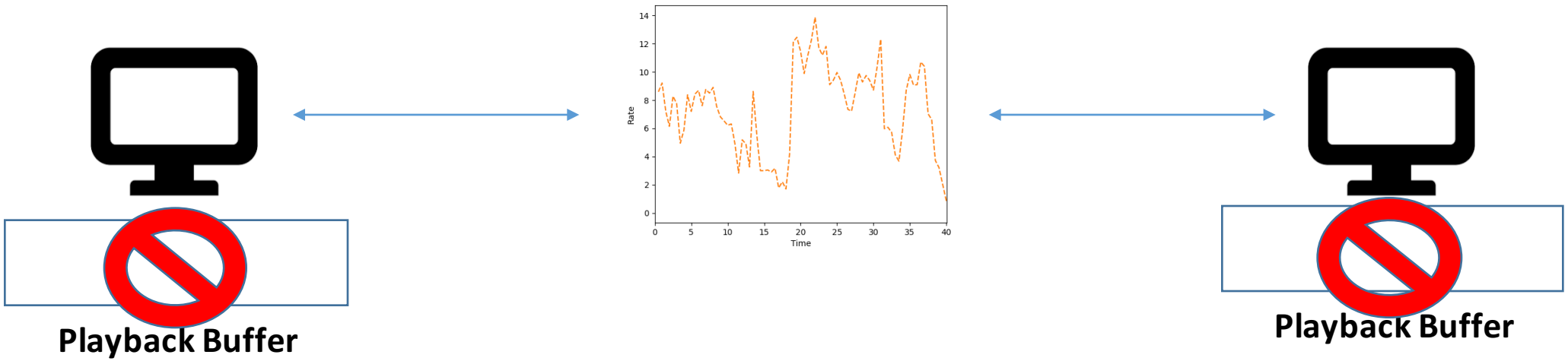
Goal: high quality, no rebuffering, smoothness

Challenges: variable/unpredictable network, conflicting QoE goals

Solution: dynamically pick chunk qualities based on QoE metric and network prediction

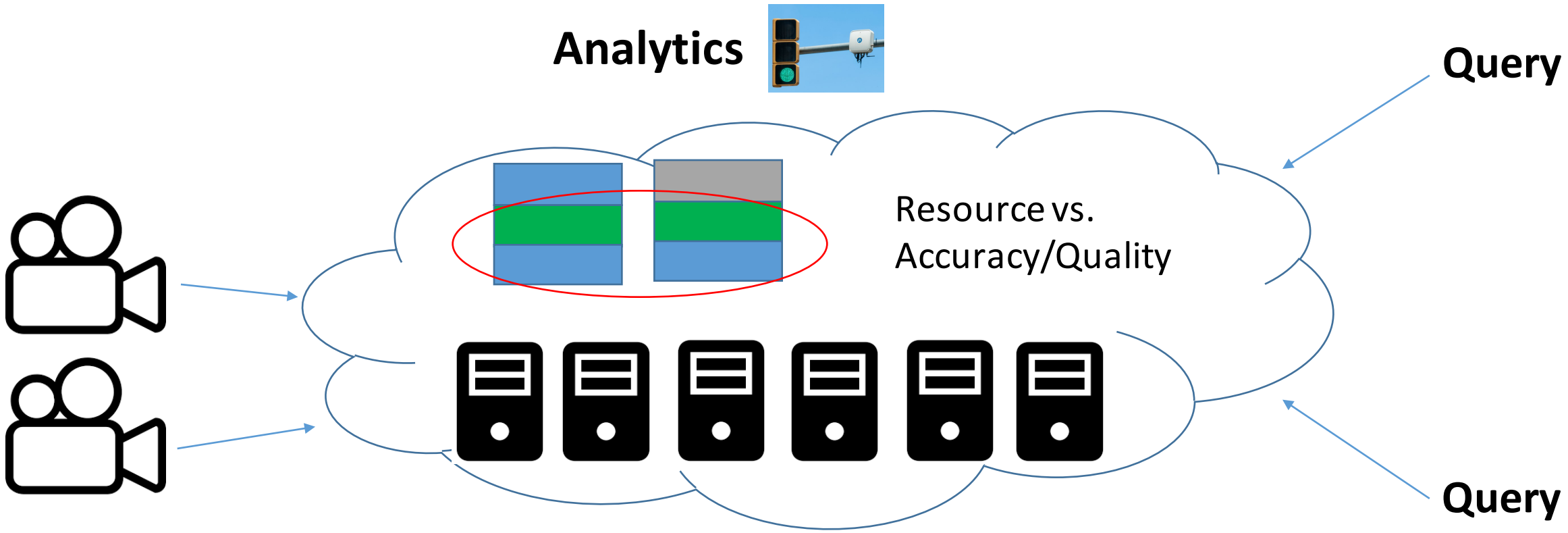
Video

Conferencing Skype



- Similar to streaming, but new real-time video, so no buffer
- **Solution**: need better predictions of network, or tighter coupling between network and video encoder

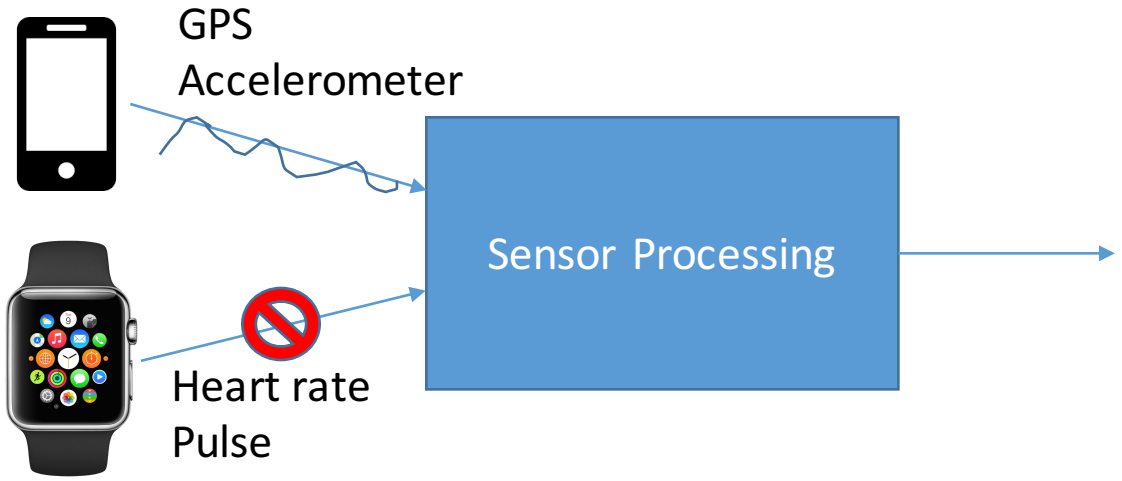
Video



- Run ML pipelines (e.g., object recognition) to answer queries on live video
- Challenges: upload bandwidth, edge compute, cluster scheduling

Mobile Systems

Sensing systems



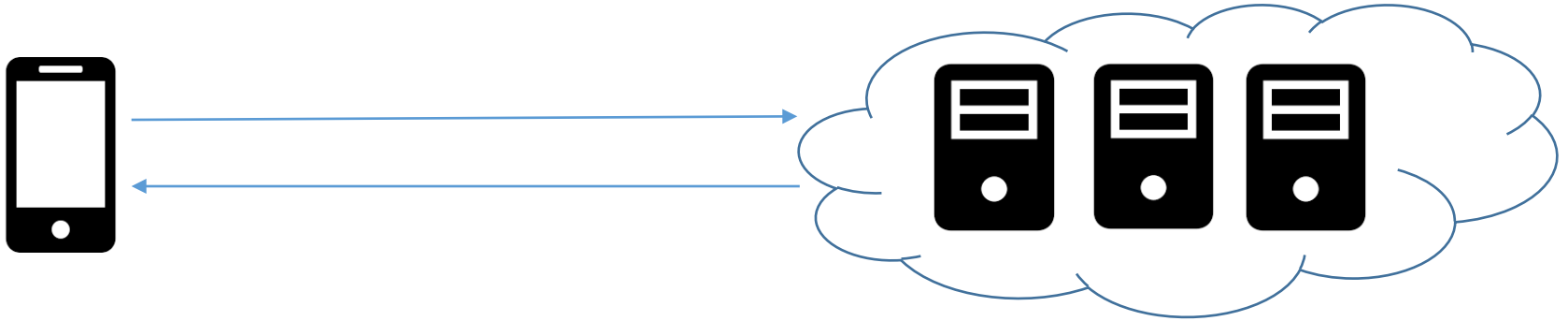
- Action tracking (e.g., walk vs. run)
- Driving monitoring
- Predicting next action (e.g., to start loading)
- Health monitors

Challenges: noisy and unreliable data, lots of sensor data, slow processing

Solution: correct data with other sensors, data history analysis (for filtering)

Mobile Systems

Offloading systems



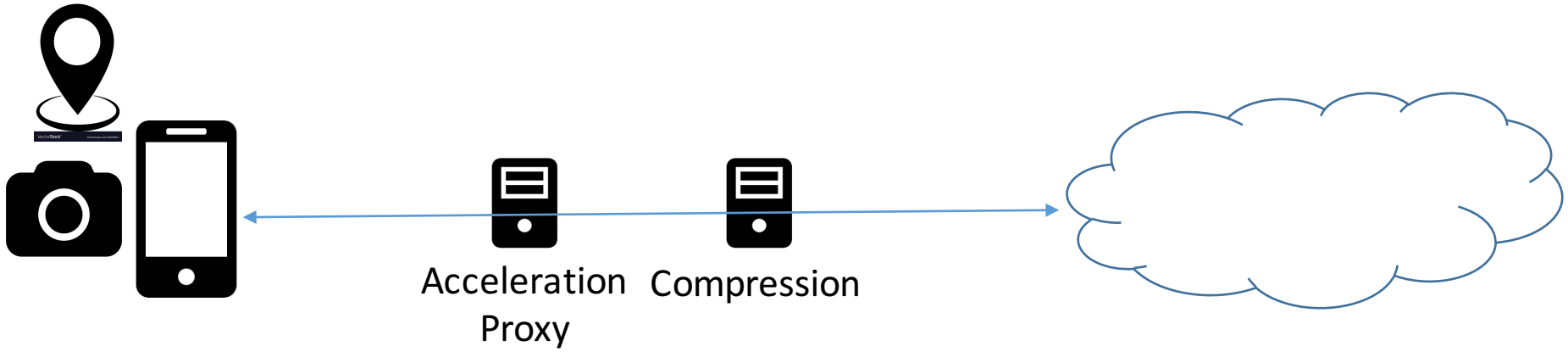
Phones <<<< Servers in terms of storage/compute → run intense tasks on servers

- Need to send state/context to server for execution

Challenge: latency to servers affects applications!

Solution: hide latency with other parts of application, subsample

Security



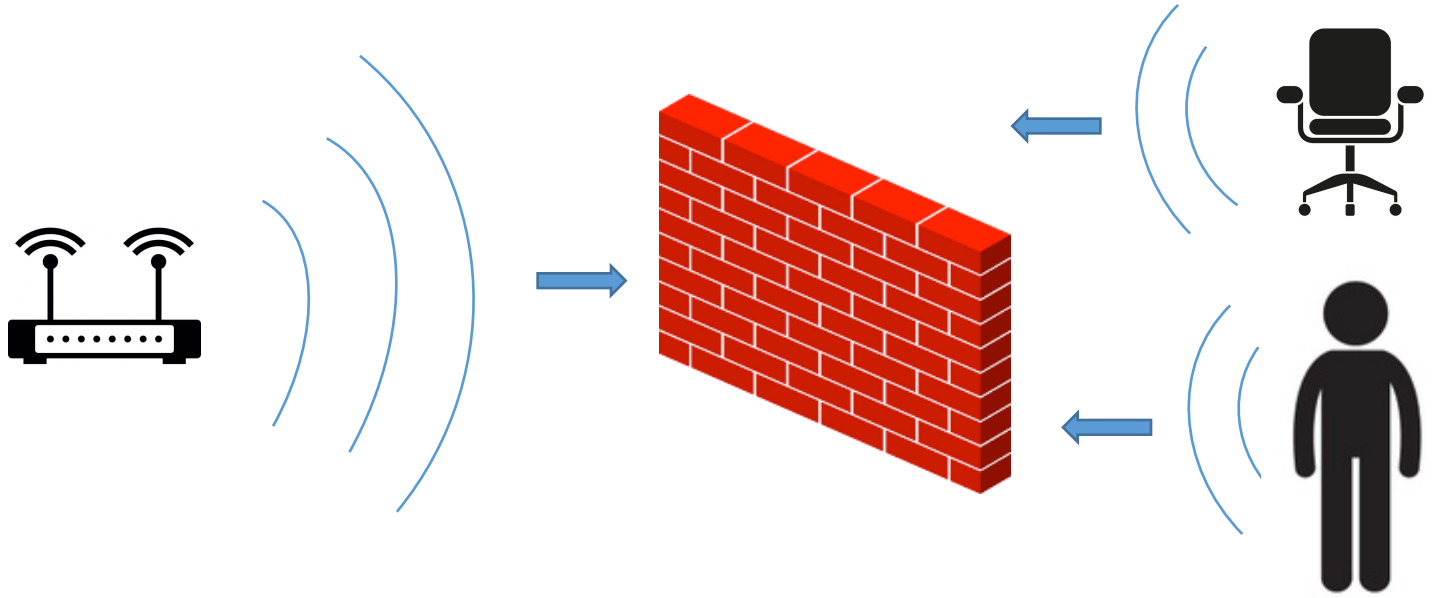
Debugging in applications (client or server)

- To provide services, apps touch sensitive data (e.g., passwords, pictures, location)
- Solution: IFC and dynamic flow control techniques to track where sensitive data flows
  - Out of device and cross domain

Network security

- Important because middleboxes are helpful!
- Challenge: prevent eavesdroppers while enabling middlebox solutions
- Solutions: end-to-end protocols and crypto solutions

Wireless Applications



- WiFi signals go through walls → track across rooms?
- Challenges:
  - Reflections from other objects
  - No context (how to associate with event?)
  - Weak signals (10 billion times weaker)

IoT Systems

Goal: connect lots of diverse devices to simply tasks and provide rich services

Challenge: lots of data, limited device power, new threat surfaces

Solution: audit transmission access, platform for data management

Something else?

- We have flexibility!
  - Last lecture before project presentations is open: send requests!
  - Open to swapping in some topics

- Other topics that may be interesting?
  - Blockchain…
  - Systems for machine learning
  - Ad services
  - Virtual/augmented reality
  - Data privacy

Course Logistics

Staff

- Instructor: Ravi Netravali
  - Assistant Professor
  - Research interests: networks/distributed systems; performance and debugging of large-scale web systems
  - Office hours: by appointment
- TA: Shaghayegh Mardani
  - PhD student in Computer Science
  - Research interests: mobile web performance; bridging gap between web and mobile apps
  - Office hours: by appointment

Course Website

CS219 Home

Course Resources

Schedule

Review Submission

CS219: Web and Mobile Systems, Fall 2019

Instructor: **Ravi Netravali**
TA: Shaghayegh Mardani
Lectures: **Tuesday/Thursday 10am-11:50am** in 5419 Boelter Hall
Office Hours

- Ravi: by appointment (ravi@cs.ucla.edu)
- TA: by appointment (lana@cs.ucla.edu)

Course Overview

Web and mobile applications are a primary way for users to access critical services and content. In addition to traditional web pages, these end-to-end applications are now used to support a wide range of services including video streaming, video conferencing, IoT tasks, machine learning pipelines (e.g., for image/video analytics), etc. This research paper-based course will study modeling analyses that highlight the burden that applications impose on systems infrastructure, and explore the systems and networking techniques that enable these end-to-end applications to remain performant, reliable, secure, and bug-free, as they support richer and more complex tasks.

Grading

- 15% Participation in paper discussions
- 5% Paper summaries
- 20% Paper presentation
- 15% Midterm exam
- 45% Final project (report and presentation)

Paper Summaries

This course will be largely based on research papers. Prior to each class, students will be expected to read the listed research paper(s) and write up a brief summary for each. Paper summaries should include the following components:

http://web.cs.ucla.edu/~ravi/CS219_F19/

Course Goals

- Learn how to read network/systems research papers critically
  - Compare similar and seemingly different papers
- Articulate understanding and thoughts about paper
- Formulate and carry out research projects
- Present research results

Who should take this course?

- Course involves a lot of research paper reading (3-4 papers per week)
  - Reading each paper will take several hours
  - Understanding the paper (and related work) will take even more time
- Not much programming (other than for research project)
- Course is mainly designed for PhD students
  - Masters students: welcome, but please note course focus
  - Undergrads: please discuss enrollment with me
- Prerequisites: knowledge of networking, OS, and distributed systems

Course Structure

- Before Class
  - Read papers
  - Submit paper critiques
- During Class
  - Paper presentations
  - **Lively** discussions
- Throughout the quarter
  - Research project
- Exam (in-class)

Paper Reading

- 1-2 papers per lecture
- Papers will be in same high-level area
  - I will highlight cases where material (e.g., background) is duplicated to reduce reading load
- "How to Read a Paper" by S. Keshav (https://blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/07/paper-reading.pdf)
  - 1st pass: high-level (title, abstract, intro, section titles); categorize paper (by area/goals), is solution plausible, etc.
  - 2nd pass: more detail (graphs/illustrations); understand main contribution
  - 3rd pass: be able to re-implement paper solution from scratch and identify flaws

Paper Reviews

- Each paper review should include:
  - Paper summary (few sentences): problem addressed, and how?
  - Potential limitations of the solution (e.g., cases where it won't work)
  - Potential extensions to make better or extend to other scenarios
  - Any questions about the paper or general topic

- **Looking for critique, not abstract only**

- You should submit a paper review for each paper (not per lecture)

- Graded on 1-5 scale (mostly on display of thought)

Paper Reviews

- Due by **10pm** the night **before** each lecture
  - Lets me identify questions that many people have
  - Important to give yourself time to think about the paper (helps discussion)

- Submit paper reviews using the form on the course website (http://web.cs.ucla.edu/~ravi/CS219_F19/review.html)

- You may skip up to 4 paper summaries without penalty

Paper Presentations

- Group of students (size TBD) will present each paper

- “Conference style” presentations
  - Domain and relevant background for the paper
  - Problem statement and challenges
  - Solution
  - Results (along with setup details)
  - Potential limitations and improvements

- Presentations should be roughly 25 minutes

Presentation Sign-ups

- Due next monday: first come first serve?
  - I will send out a spreadsheet later today
- Drop policy: the show must go on!
- Incentives for early presentations
  - Expectations are less defined
  - Less interference with research project
  - Less overlap with work for other courses

Paper Discussion

- Presenters lead the discussion
  - **But everyone should participate**

- Come prepared with:
  - Questions to discuss
  - Comparison points of the two papers
  - Discussion of key takeaways from paper
  - Discussion points for limitations
  - Potential extensions (good time to get feedback on ideas!)

- Presenters for a given class should coordinate for discussion

Research Project

- Topic: web/mobile systems are broad--anything related should be fine

- **Aim high!**
  - Goal: design/evaluate new research idea or extension to one
  - Goal is *not* to re-implement something
  - Hope some projects turn into longer research projects

- Groups of 2-3 (depending on enrollment)

Research Project Timeline/Deliverables

- **Project proposal** (2-3 pages): describe high-level problem you are tackling, related work, proposed solution, and implementation plan
  - Okay to pivot!
- **Project presentations**: 20 minute in-class presentations (conference style)
- **Final report** (6 pages): conference-style paper detailing problem, challenges, solution, results, and related work

Project Notes

- Using existing research projects should be fine, but let's discuss
  - Remember that projects are still done in groups
- **Please start thinking early!**
  - I'm happy to discuss project ideas before official date
- Example Ideas
  - Video streaming algorithms for specific networks
  - Automatic selection of web optimization strategy
  - Security policies in web-to-app converters

Exam

- In class on Tuesday, November 19
- Short answer about different application scenarios
- Goal: devise solutions using techniques we discussed (or others)
  - Important to understand tradeoffs of the solutions
- Many answers are correct; thought process is most important part (perfect solution may not exist)

Grading

- 15% Participation in paper discussions
- 5% Paper summaries
- 20% Paper presentation(s)
- 15% Exam
- 50% Final project (report and presentation)

Other Notes

- Please drop early
  - Affects paper presentations and research projects
- Any issues (e.g., critique deadlines, project concerns, etc.) → please come see me early
- Not a lecture course!!
  - Please come prepared and participate so everyone can benefit

For Next Lecture

- Topic: Web performance and optimizations
- Presenter: Ravi
- Papers:
  - **Polaris** (NSDI 2016): dependency-aware client-side scheduling
  - **Vroom** (SIGCOMM 2017): push/preload policies to send content ahead of time
  - WebGaze (NSDI 2017): track user eye movement to identify important content
  - Vesper (NSDI 2018): time-to-interactive

Any Questions?