Introduction

CS 571: Operating Systems (Spring 2022)

Lecture 1

Yue Cheng

Some material taken/derived from:

- Illinois CS-523 materials created by Tianyin Xu.

Who am I?

Yue Cheng

- Web: cs.gmu.edu/~yuecheng
- Email: yuecheng@gmu.edu

Working on systems

- Distributed systems, cloud computing, operating systems

Who am I?

Yue Cheng

- Web: cs.gmu.edu/~yuecheng
- Email: yuecheng@gmu.edu

Working on systems

- Distributed systems, cloud computing, operating systems

- Happy to chat about industry vs. academia

Our GTA: Yuyang Leng

2nd year PhD student

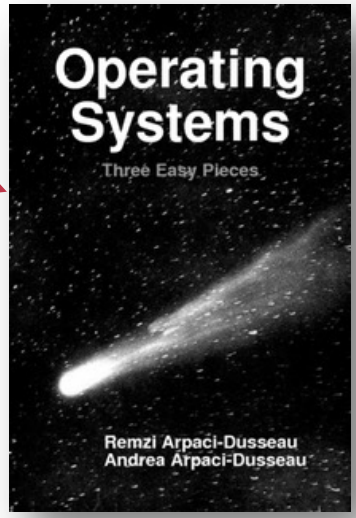
- Email: yleng2@gmu.edu
- Office hours:
  - Tuesday and Thursday, 5-6pm, BB Collaborate
- Research interests:
  - Deep learning in Internet of things (IoT) and cyber-physical systems (CPS)

Administrivia

No required textbook

Recommended textbook

- Operating Systems: Three Easy Pieces (OSTEP)
  By Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau

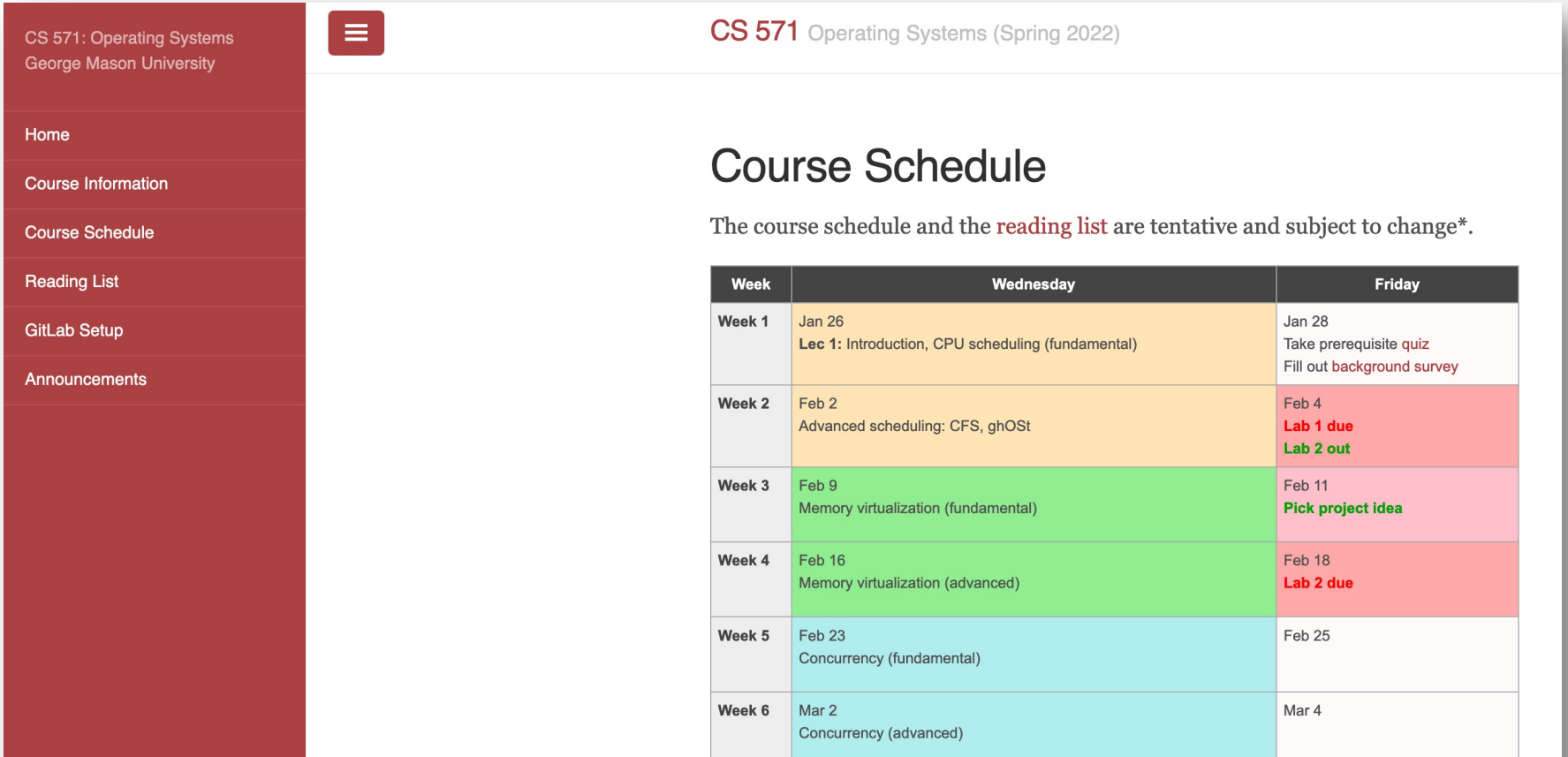


Prerequisites are enforced!

- CS 310 Data Structures
- CS 367 Computer Systems & Programming
- CS 465 Computer Systems Architecture
- Be comfortable with C programming language

Course web page

- https://tddg.github.io/cs571-spring22/
- Class materials will all be available on class page



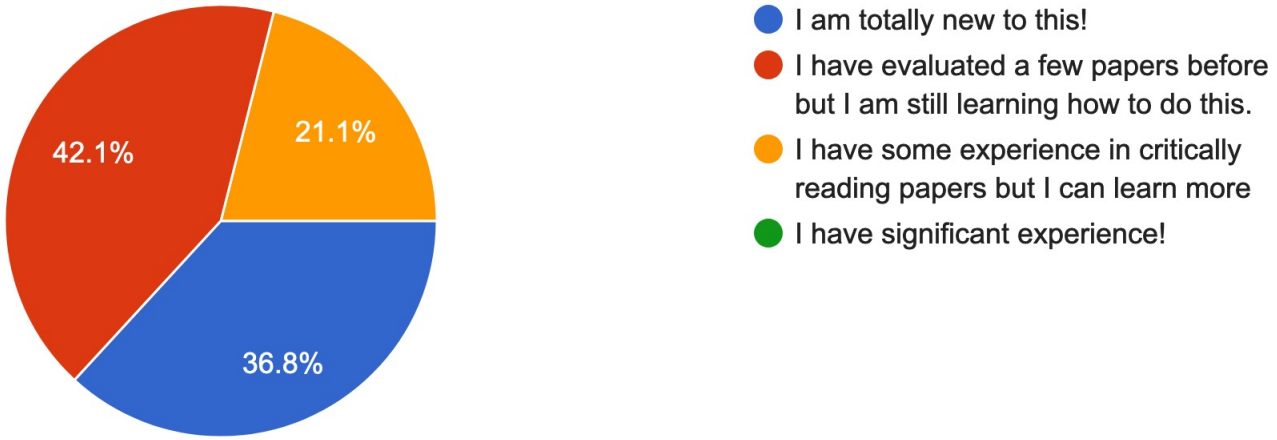
Online forum

- We will be using Ed
- Please enroll in Ed by Week 1 via the link I broadcast thru BB

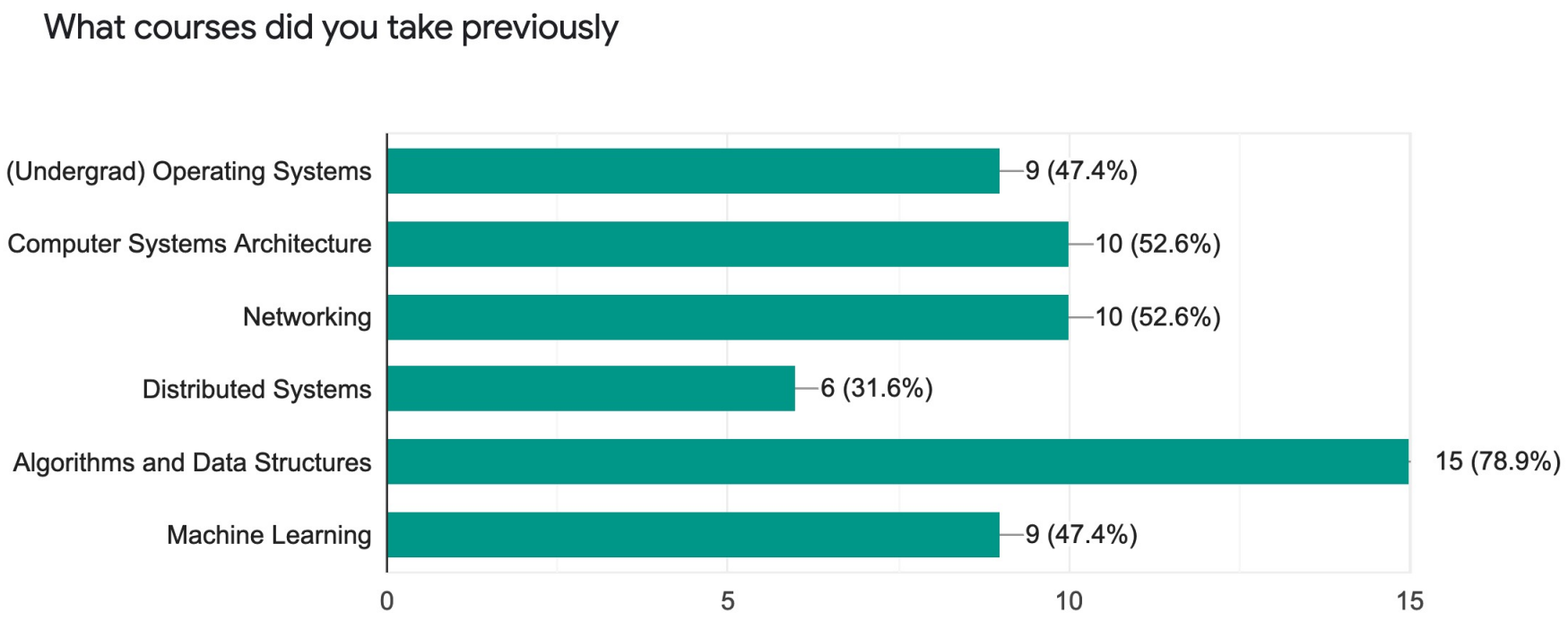
Background survey

Paper reading

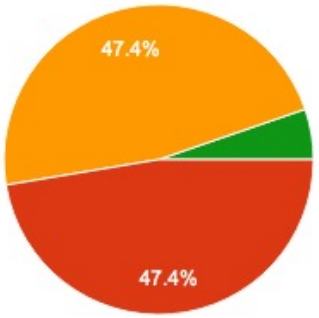
How much experience do you have in critically reading and evaluating systems research papers?

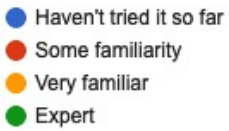


Prior courses

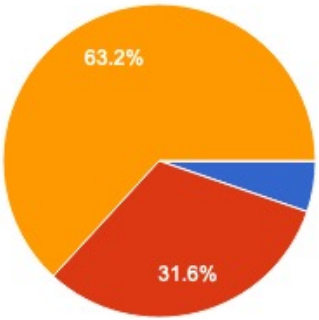


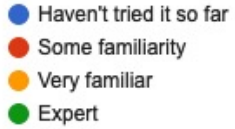
Familiarity with tools



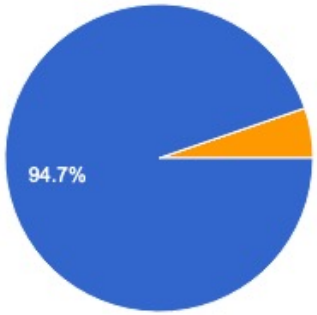


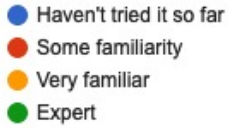
C/C++



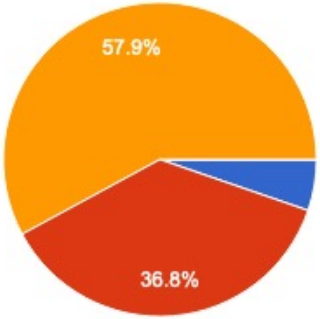


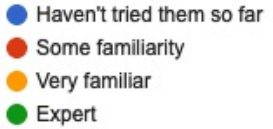
Java



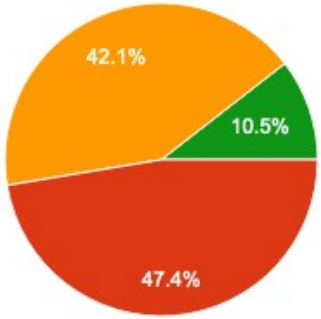


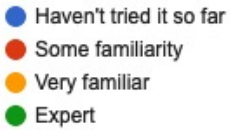
Go





Linux bash/
cmd-line tools





Python

What do you hope to learn from this course?

I really love learning about low-level systems, so I hope to take away as much knowledge about the inner mechanisms of low-level systems as I can from this course. This will be very useful for me both in my career and upcoming graduate courses.

A much deeper understanding of how operating systems work and the current research on system optimization and other interesting topics. I am most excited about learning new concepts that I never even knew existed within operating systems and how it will shape how I build systems in the future.

My interest down the road is about distributed systems and parallelism, CS 571 is a prerequisite for both of those topics. I hope to have a good understand of OS before tackling the topics in interest above.

I'm specifically interested in learning about persistent, scalable data stores and how to make them more effective and efficient.

Looking for more understanding of OSs in support of my dissertation on security in 5G non-terrestrial networks.

…

What is this course about?

- Half OS fundamentals, half systems research
- Covered fundamentals provide direct support for research topics to be discussed

What is this course about?

- Half OS fundamentals, half systems research
- Covered fundamentals provide direct support for research topics to be discussed

- More importantly, it's about **systems research**
  - To develop a systematic understanding of systems research
  - To discuss seminal and recent systems research
  - To get hands dirty in systems research
    (via a term-long research project)

This course does not teach

- Lots of basic OS concepts (those listed in the prerequisite quiz)
- Hands-on skills of hacking an OS kernel
  - Linux hacking experience not required for 571
  - Systems research is much broader than OS kernel
- 471 is the choice if you want to learn the above
- Take the prerequisite quiz on the course webpage for a quick self-assessment

Big topics covered by this course

- Virtualization: CPU and memory
- Concurrency
- Persistence
- VMs/containers
- Distributed systems

Course format

- For (most of) the topics
  - One week for fundamentals
  - One week for paper discussion
- Fundamental sessions delivered in lectures
  - Mostly I talk
- Paper discussion sessions
  - Paper review + in-class discussion
  - I will ask lots of questions

You are expected to

- Attend all fundamentals+discussion sessions
- Read the assigned papers + actively engage in discussion (15% paper review + 15% participation)
- Conduct a term-long research project (3x10%+40%)

- Reminders
  - Honor code
  - Late policy: 15% deducted each day. No credit after 3 days

What are the target students?

- Students who are actively doing (systems-related) research
  - Review seminal and recent systems works
  - Explore and discuss new ideas
- Students who are interested in systems research
  - Evaluate if systems research is something for you

- If you are neither of the above, you may reconsider

But, how to read a research paper?

- Paper reading is an important skill in grad school
  - To learn how to efficiently and effectively read a systems research paper
  - Might be slow at the very beginning
  - But you will get better the more you practice reading

- We will read 10+ papers in thorough detail
  - Some of them are must-read, seminal papers (MapReduce)
  - Some are recent works of trending topics (learned index)
  - Impossible to innovate if you don't know the literature well

#1 rule:

Do NOT worship any paper or author

- A paper is not a "truth" but an "opinion" (or claim)
  - You should have your own judgement

- Critical thinking is a must in grad school
  - Papers are arguments based on authors' work / perspective
  - You are welcome to reject the arguments, criticize the approaches, and question the results
  - However, you will need to backup your criticisms and rejections

#1 rule:

Do NOT worship any paper or author

- A paper is not a "truth" but an "opinion" (or claim)
  - You should have your own judgement
- Critical thinking is a must in grad school
  - Papers are arguments based on authors' work / perspective
  - You are welcome to reject the arguments, criticize the approaches, and question the results
  - However, you will need to backup your criticisms and rejections
- There are horrible papers published at top venues
  - You need a legit reason to "attack"

How to read a research paper?
(Griswold's version)

1. What are the motivations?
2. What is the proposed solution?
3. What is the work's evaluation of the proposed solution?
4. What is your analysis of the identified problem, idea, and evaluation?
5. What are the contributions?
6. What are future directions for this research?
7. What questions are you left with?
8. What is your takeaway message from this paper?

https://cseweb.ucsd.edu/~wgg/CSE210/howtoread.html

How to read a research paper? (Cheng's version)

1. What problem does this paper tackle?
   - Why is it important or why do you think it is not as important as the authors claim (do you disagree)?
2. What is the solution design and implementation?
   - What is the limitation of the solution?
3. Does it resonate your experience or benefit your own research or work?
   - If so, how do you want to use their solution?
4. What do you learn from this paper?

Paper discussion

- You are expected to read the assigned papers before the class
  - 1-2 papers for each paper discussion session
  - Finish the paper reviews (posted several days before class)
  - Do not come to class if you don't read :-)

- We will discuss the papers in class

Paper discussion (cont.)

- I may provide slides (and/or use original paper pdf) to facilitate discussion

- We will discuss questions by cold calls
  - I will prepare a list of questions
  - You can also ask questions

- You may volunteer to answer questions, or I will randomly ask students to discuss questions
  - You will either be embarrassed if you are not prepared or pretend you were not in class (and lose points)

Research projects

- You will conduct a research project fitting in the broad definition of "computer systems"
  - In a team of 1 or 2 students
  - If you have strong reasons to do a large one in a team > 3, come talk to us first

- I have already posted a list of ideas
  - You are expected to form teams by end of Week 2
  - By filling a team composition Google form

Project timeline (13 weeks)

- End of Week 2: Form a team
- End of Week 3: Pick a project idea

Project timeline (13 weeks)

- End of Week 2: Form a team
- End of Week 3: Pick a project idea
- End of Week 5: Submit a project proposal
  - A well-defined research problem and proposed solution
  - Feasibility of your proposed solution (via examples, tools)
  - Use me as a sounding board

Project timeline (13 weeks)

- End of Week 2: Form a team
- End of Week 3: Pick a project idea
- End of Week 5: Submit a project proposal
  - A well-defined research problem and proposed solution
  - Feasibility of your proposed solution (via examples, tools)
  - Use me as a sounding board
- End of Week 9: Submit checkpoint 1 report
  - Show preliminary results and design and (partial) implementation of your prototype (via simple examples)

Project timeline (13 weeks)

- End of Week 2: Form a team
- End of Week 3: Pick a project idea
- End of Week 5: Submit a project proposal
  - A well-defined research problem and proposed solution
  - Feasibility of your proposed solution (via examples, tools)
  - Use me as a sounding board
- End of Week 9: Submit checkpoint 1 report
  - Show preliminary results and design and (partial) implementation of your prototype (via simple examples)
- End of Week 12: Submit checkpoint 2 report
  - At this point you are expected to have built a solution and get ready for evaluation
  - Describe detailed evaluation plan

Project timeline (13 weeks)

- End of Week 2: Form a team
- End of Week 3: Pick a project idea
- End of Week 5: Submit a project proposal
  - A well-defined research problem and proposed solution
  - Feasibility of your proposed solution (via examples, tools)
  - Use me as a sounding board
- End of Week 9: Submit checkpoint 1 report
  - Show preliminary results and design and (partial) implementation of your prototype (via simple examples)
- End of Week 12: Submit checkpoint 2 report
  - At this point you are expected to have built a solution and get ready for evaluation
  - Describe detailed evaluation plan
- Week 15: Final project demo (10 mins)
- Week 16: Submit final project report (6 pages)

Something interesting this semester!

Open studio (Friday 10-11:30am)

- Discuss and debate your research ideas and proposed solutions
  - You are encouraged to prepare just a few slides
  - Email me beforehand if you would like to present
  - Everyone welcome to attend (not required though)
- Get to learn what other teams are working on
- Brainstorming and getting feedback

Short project presentations

- You are more than welcome to present your work (in a more formal way) in front of all at the beginning of each class
  - To seek feedback
  - To articulate idea, design, implementation
  - To share insights and findings

- 10-20 minutes should be fine
  - Need pre-arrangement by letting me know before the class

Project ideas

- Well connected to OS, distributed systems, and ML systems
- I provide a list of ideas which I think are interesting (or important)
  - My goal is to get an interesting distribution of projects
  - Yours is to do a project you like most
  - Advice: Don't get too attached to any one project; rather, have a few in mind that would be palatable

Project ideas

- Well connected to OS, distributed systems, and ML systems
- I provide a list of ideas which I think are interesting (or important)
  - My goal is to get an interesting distribution of projects
  - Yours is to do a project you like most
  - Advice: Don't get too attached to any one project; rather, have a few in mind that would be palatable
- Of course, you are welcome to reuse your own research project
  - Need to discuss with me about whether it falls into any systems project category

Project categories

- New system
  - Design and implement a new system with new properties and capabilities

Project categories

- New system
  - Design and implement a new system with new properties and capabilities
- New tool
  - Design and implement a new tool that provides new functions

Project categories

- New system
  - Design and implement a new system with new properties and capabilities
- New tool
  - Design and implement a new tool that provides new functions
- Measurement
  - Experimentally measure some metrics of a system

Project categories

- New system
  - Design and implement a new system with new properties and capabilities
- New tool
  - Design and implement a new tool that provides new functions
- Measurement
  - Experimentally measure some metrics of a system
- Study
  - Qualitatively or quantitatively analyze a type of systems

Evaluation of your research projects

- Evaluation criteria
  - Overall merit
  - Why the problem is important
  - Originality and insightfulness
  - Validation and thoroughness
  - Presentation and clarity

Project grading

- Excellent (A to A+): significant results and publishable work
- Very good (A- to A): strong results and a clear roadmap towards publishable work
- Good (B+ to A-): interesting results but quite far from being significant
- Fine (B to B+): a good exploration but leads to nothing
- Well, OK… (B- to B): some efforts of exploration with no conclusion

Project grading

- Excellent (A to A+): significant results and publishable work
- Very good (A- to A): strong results and a clear roadmap towards publishable work
- Good (B+ to A-): interesting results but quite far from being significant
- Fine (B to B+): a good exploration but leads to nothing
- Well, OK… (B- to B): some efforts of exploration with no conclusion

You should have the courage to explore and fail

Systems research conferences

- SOSP/OSDI (one conf w/ two names)
- NSDI (networked systems)
- FAST (file and storage systems)
- EuroSys (systems venue held in Europe)
- USENIX ATC (everything related to systems)

- ASPLOS (architecture + PL + systems)
- SoCC (cloud systems)
- SC (everything about HPC + parallel computing)

http://csrankings.org

Systems research conferences

Project turnaround time is way longer

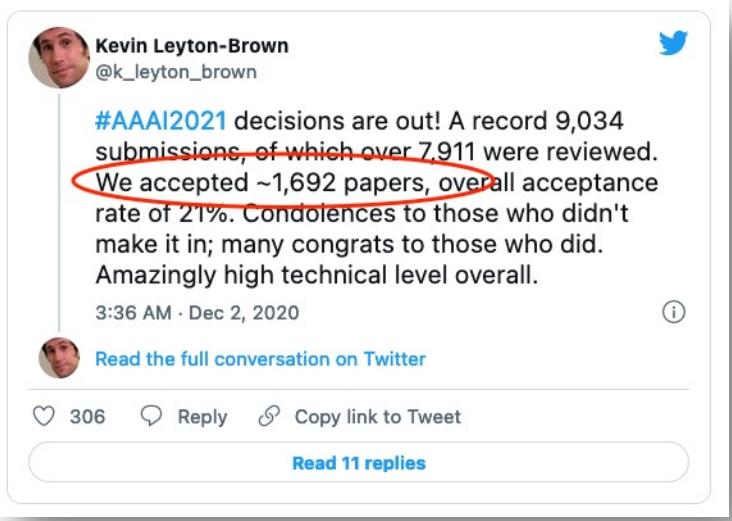
So as the publication cycle

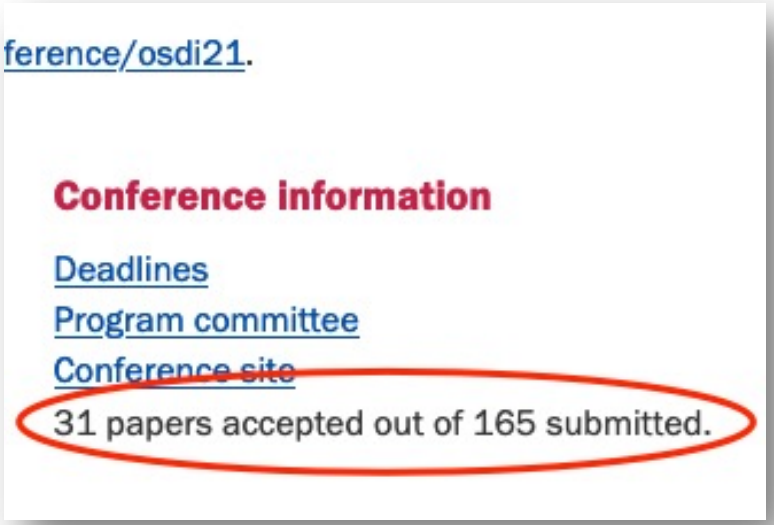
In this section, we evaluate on AWS Lambda.
Implementation. We have implemented a prototype of atop by adding 17,756 lines of Go code using less than two person-years: 639 LoC for the client library, 7,678 for the client daemon, 5,962 for the Lambda runtime, and 3,477 for utilities shared across components [14].

Systems research conferences

Project turnaround time is way longer

So as the publication cycle





Not many papers to read, but you are expected to read just the small amount of published work

Useful tips (I)

- How to find good research ideas?
  - Identify important problems
  - Problem first!

Useful tips (I)

- How to find good research ideas?
  - Identify important problems
  - Problem first!

- How to identify important problems?
  - From your own experience
  - Extensive reading

Useful tips (I)

- How to find good research ideas?
  - Identify important problems
  - Problem first!

- How to identify important problems?
  - From your own experience
  - Extensive reading

  Many works aim to solve same problem, e.g., mitigating request tail latency, minimizing cloud functions' cold start penalty

These are the important problems, typically

Useful tips (II)

- Talking with others may be helpful, sometimes
  - Articulate the problem that you want to solve
  - Pitch your idea to peers/experts

Useful tips (II)

- Talking with others may be helpful, sometimes
  - Articulate the problem that you want to solve
  - Pitch your idea to peers/experts

- Caveat: Think like a novice

Experts often make too much abstraction

  - *"This has been done two decades ago…"*
  - *"Isn't this the same as…"*

Novice: Excitement and Courage

Useful tips (III)

- Understanding the problem first!
  - Innovation w/o understanding leads to BS

Useful tips (III)

- Understanding the problem first!
  - Innovation w/o understanding leads to BS
- Measure, then build *
  - Empirical studies and measurements are great ways to develop understanding
    https://www.usenix.org/conference/atc19/presentation/keynote
- If you have a general topic/direction/problem but don't have a crisp idea, work on a study or a measurement
  - Ask yourself questions and let curiosity guide you

Useful tips (IV)

Novelty != Never done before

Useful tips (IV)

Novelty != Never done before

Example:

Log-structured merge-tree (LSM-tree), a 1993 patent →

Circa 2003, Google discovered this patent and built SSTable (core data format of Google BigTable) and LevelDB on the patented LSM-tree idea

Useful tips (V)

- New driving forces can bring innovation
  - New scale: warehouse datacenters, planet-scale, geographically distributed datacenters
  - Unique applications: MapReduce, Dynamo, TensorFlow
  - Emerging computing paradigm: Serverless computing
  - New hardware: non-volatile memory
  - …

Find your teammate + explore project ideas, now

- Ed (the "Search for teammate" section)
  - I'm Yue Cheng, an Nth year grad student
  - I'm interested in operating systems
  - I pan to build a new Datacenter OS that does …
- Form a team due in 1 week
- Pick an idea due in 2 weeks
- Project proposals due in 4 weeks (one page)
  - What do you plan to do?
  - Why is it interesting?
  - How will you do it (feasibility)?
    - Basic idea
  - What's your plan and schedule?
  - What resources do you need?

FAQ

- I have no interested in research. Should I drop?
  - If you are interested in systems in general, then see if any excites you
- I don't want to come to open studio. Can I skip?
  - Yes
- How can I get feedback?
  - Chat with me
  - Attend (or present at) the open studio and brainstorm with peers
  - Present at the beginning of a class to get feedback all around