Advanced Operating Systems (CS 523)

Fall 2019

Tianyin Xu

meta-info

- Instructor: **Tianyin Xu**
- Teaching assistant: **Jack Chen**
- Course website (with course schedule)
  - https://cs523-uiuc.github.io/fall19/
- Slack:
  - #cs523 at the SysNet slack
- Piazza (**do we need it?**)

$ whoami

- Assistant Professor in the CS department
- Working on software and system reliability
- Worked at Facebook on dealing with datacenter-level failures before joining UIUC
  - projects completely different from my PhD thesis
  - happy to chat about industry vs. academia
- Did grad school at UC San Diego.
  - That's a dream job: I wish I could be a tenured grad student.
- Applied twice for grad school.
  - I failed the first time.
  - persistence >> genius

What is this course about?

- **It's all about ~~Operating~~ System Research**
  - Develop a systematic understanding of *system research*
  - Grasp the basic knowledge of *system research*
  - Discuss the seminal *system research* papers
  - Get feet wet in *systems research* (mini research project)

- **This is a course about:**

  "discussing research" + "doing research"

This course does NOT teach:

- Basic concepts of operating systems
- The skills of hacking an operating system kernel
  - Kernel hacking experience is not required for 523.
  - System research is much broader than OS kernel.
- **CS 423** is the choice if you want to learn the above.
- There is a prerequisite quiz on the course website.

Who are the target students?

- Students who are actively doing systems research.
  - Review classic, seminal papers
  - Discover and discuss new ideas
  - Try out new ideas
- Students who are interested in system research.
  - Evaluate if systems research is something for you.

- If you are neither of the above, you need to be aware that this course is not designed for you.

You are expected to:

- **Read research papers (before the class):**
  - 2 papers for each class
  - Do not come to class if you don't read ☺
- **Discuss the reading (in class)**
  - Again, it's hard to discuss if you don't read.
- **Conduct a semester-long mini-research project**
  - The best way to learn is to do it
  - The main purpose of the course -- **85% of your final grade**

No midterm or final exam!

Reading

- **Reading papers is one of the most important skill sets in grad school.**
  - You need to learn how to **efficiently** and **effectively** read research papers
- **We will read a lot of classic papers.**
  - Those are the must-to-read papers for systems research.
  - Some of them will appear in the SysNet qual exam.
  - You can't innovate if you don't understand them deeply.
- **Come to CS 591 SN if you want to discuss new papers.**

Do NOT worship any paper or author

- **A paper is not a "truth" but an "opinion"**
  - You should have your own judgement
- **Critical thinking is a must in grad school**
  - Papers are arguments based upon research.
  - You are welcome to reject the arguments, criticize the approaches, and question the results.
  - You will need to back up your criticisms and rejections.
- **There are plenty of horrible papers published in top conferences.**
  - But you need a legit reason to "attack."

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

1. What are the **motivations** for this work?
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 take-away message from this paper?

W. Griswold, How to Read an Engineering Research Paper, https://cseweb.ucsd.edu/~wgg/CSE210/howtoread.html

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

1. What **problem** is the paper solving?
   - Is it a real problem or a fake/imaginary problem?
   - Is it an important problem? What's the consequence if the problem is not solved?
   - How prevalent is the problem? How many people can benefit from a solution?
2. Does the proposed solution **practically** solve the problem?
   - If not, how much it *actually* solves?
   - Do you believe in the solution?
   - Do you want to use the proposed tool/system?
3. Do you **like** or **hate** the paper? Why?
4. What do you **learn** from the paper?

Topics we will be discussing

- Historical Perspectives
- Unix and Plan 9 (and MINIX and Linux)
- Microkernel
- Library OS
- Synchronization
- Scheduling
- Memory Management
- Virtualization
- Storage and File Systems
- Communication
- Distribution
- Protection
- Reliability

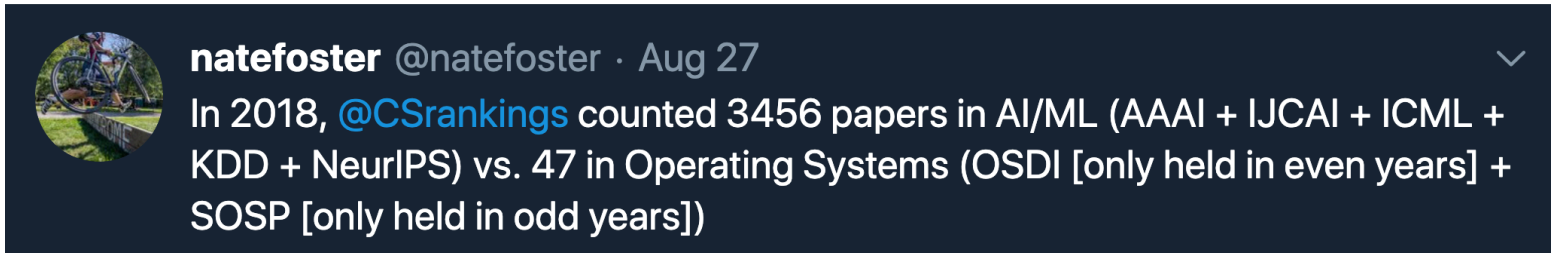
System research conferences

- **SOSP/OSDI** (one conference with two names)
- ASPLOS (PL and arch)
- NSDI (networked systems)
- FAST (file and storage systems)
- EuroSys (European)
- SOCC* (Cloud systems)
- Sigmetrics* (measurement)
- USENIX ATC* (everything)
- (related) ICSE/FSE, CHI, MobiSys/MobiCom, SIGCOMM*, IMC, PLDI*

* I never (or failed to) publish there.

System research conferences

- The research cycle is long.
- So as the publication cycle.



- There not many papers to read, but you are expected to read the small number of published work.

Class Discussion

- **There will be no “lecture.”**
  - This is a 500-level course.
- **We will discuss papers by playing card games.**
  - We will practice it today ☺
- **You can volunteer or will be asked to discuss questions in class.**
  - If you do not read the paper, you will be embarrassed.

Course Project

- **A research project fitting in the broad definition of "systems."**
  - In a group of **1** or **2**.
  - If you have strong reasons to do a **large** project in a team of more than 2, talk to us first.
- **Please form groups before the end of next week.**
  - Send me an email by the end of next week identifying who is in your group

Most projects fall into the following categories:

- **Study**: qualitatively or quantitatively analyze an important aspect of one type of systems.
- **Measure**: measure and characterize an important aspect of one type of systems through experimentations.
- **Tool**: design and implement a new tool that can address an important problem in modern systems
- **System**: design and implement a novel system with new capabilities or properties

Examples

- **Study**
  - Chou et al., An Empirical Study of Operating Systems Errors, SOSP 2001.
- **Measure**
  - Pillai et al., All File Systems Are Not Created Equal: On the Complexity of Crafting Crash-Consistent Applications, OSDI 2014.
- **Tool**
  - Li et al., CP-Miner: A Tool for Finding Copy-paste and Related Bugs in Operating System Code, OSDI 2004.
- **System**
  - Rosenblum et al., The Design and Implementation of a Log-Structured File System, SOSP 1991.

Evaluation of research projects

- **It will be evaluated using the same criteria as SOSP/OSDI submissions.**
  - **Overall merit**
  - Importance of the topic
  - Originality and insightfulness
  - Validation and thoroughness
  - Presentation and clarify

- **Dream bar**: CP-miner, Veriflow
- **High bar**: sufficiently interesting to be a real paper
- **Low bar**: something you can brag about

What if you are not able to (or not interested in) find decent idea?

- **We prepared a measurement project for you.**
- **Container measurement project (CMP)**
  - Measure the performance overhead of modern container techniques (e.g., Docker versus gVisor)
  - Understand the nitty-gritty details that affect the performance of containerized applications
  - Learn how to conduct a solid measurement study
    - which will benefit your future research/work
  - Potentially aiming for a sigmetrics-like conference
  - Link: https://cs523-uiuc.github.io/fall19/cmp.html

Project Timeline (12 Weeks in Total)

- **End of Week 3: Submit project proposal**
  - A well-defined research problem and feasible solutions.
  - Show the feasibility by concrete examples, datasets, and tools for system building.
  - **I encourage you try me the idea before deciding on the project.**
- **End of Week 7: Submit Checkpoint 1 report**
  - Show your system/tool prototype and preliminary results.
  - Your prototypes should be able to work with your motivating examples.
- **End of Week 11: Submit Checkpoint 2 report**
  - (At this point, you are expected to build your system/tool and start evaluation)
  - Describe the detailed evaluation plan in your report.
- **Final project demo (15 min)**
- **Submit final project report (6 pages)**

Exploring your project NOW!

- **Initial project proposals due in 3 weeks (one page)**
  - What do you plan to do?
  - Why is it interesting?
  - How you'll do it (feasibility)?
  - What you're not sure about?

} **Problem Statement**

Grading

- **A to A+:** significant results and publishable work;
- **A- to A:** strong results and a clear roadmap towards publishable work;
- **B+ to A-:** interesting results but quite far from being significant;
- **B to B+:** a good exploration but leads to nothing;
- **B- to B**: some efforts of exploration; no conclusion.

(You should have the courage to explore and fail)

Tips

- **Pick a good problem**
  - Why is this problem interesting?
  - What is the impact of solving this problem?
  - Look at what others are doing:
    - Academic conferences: OSDI/SOSP, NSDI, EuroSys, SOCC, ATC
    - Engineering blogs and postmortems
- **Pick a problem that is achievable.**
  - Start from small (you only have one semester)
  - What resources would you need to investigate the problem? (ask if you're serious)
- **Think about how to evaluate your work**

We take a very broad and inclusive view of system research.

- It is well connected to areas like architecture, PL, SE, HPC, networking, and embedded/mobile.
- Security and reliability are essential aspects of system design and implementation.
- Everyone is talking about Sys4ML and ML4Sys.
- It can be even broader, e.g.,
  - Visualizing large-scale system data (e.g., logs and traces)
  - Human factors in system operations and management
  - Education of operating systems
  - Cryto for OS

Questions about the project?

- We are always here to help (use us well; but don't abuse us)
  - Slack channel
  - Email
  - Office Hour
  - Appointment

- Systems research requires no genius.
  - It requires understanding and experiences.

Let’s play the card game.

- **Write on the card**
  - Your full name
  - The name you want me to call you
  - A description about yourself

- Tianyin Xu
- Tianyin/TY/t
- A watchman in a cornfield

Let's play the card game.

- What's your name?
- What are you working on?
- What are you looking for in cs523?
- What is the coolest thing you did in the summer?

- My name is Tianyin.
- I work on system reliability.
- I hope to have fun in cs523.
- I worked with Dimitrios, Apo, and Josep on new page table designs in the summer.

Let's play the card game.

- What is an operating system?
- What is inside an operating system?
  - Is Window systems a part of OS?
    - The question discussed in the supreme court in 1998
  - How about web browser?
  - Apache web server?
  - Compiler?
  - Device firmware?

Let’s play the card game.

- What drives an OS design to change?
  - Hardware trend?
  - Application demand?
  - User demand?
- Can you give concrete examples?

Let's play the card game.

- What is virtual memory?
- Why do we need virtual memory?
- How does virtual memory actually work?

Let’s play the card game.

- What is a “virtual machine”?
- Is container a virtual machine?

Let’s play the card game.

- What is a system call?