Welcome to EECS 582

Baris Kasikci

Current Enrollment

- Enrolled: 43
- Waiting Permissions: 13
- Love the Interest in Systems!
  - *Unsustainable class size* ☹

EECS 598: Topics in Hardware Security

Instructor: Daniel Genkin, genkin@umich.edu

The security of a system is only as good as its weakest link. Even if a system's software is perfectly secure, the complex interactions between the system's hardware and the physical world have not been properly understood. Side-channel attacks exploit unintentional, abstraction-defying leakage from physical devices (such as the device's power consumption, electromagnetic radiation or execution timing variations) to recover otherwise-unavailable secret information.

In this class, we shall review recent papers in the area of side channel attacks and their mitigations. Specific topics include (but not limited to):

1. Physical side channel attacks such as power and electromagnetic analysis
2. Microarchitectural attacks such as cache attacks, and Rowhammer
3. Speculative execution attacks: Spectre, Meltdown and Foreshadow
4. Side channel mitigations and countermeasures

Class requirements:

1. 45min – 1hour presentation
2. Final project
3. Active participation in paper discussion

Class prerequisites: Prior experience in low level programing (C / C++ / assembly) is required. Familiarly with basic signal processing (for physical attacks) as well as basic operating system principles (for microarchitectural attacks) will be helpful. The class might also include some basic cryptographic background which is required for understanding attacks on cryptographic systems.

Disclaimer

- See paper/image references on individual slides
- Some content borrowed from:
  - *Mosharaf Chowdhury's EECS 582 Lecture Slides*
  - *Steve Hand's advice on How to do a Systems PhD?*
  - *George Candea's Know Thy Neighbor Talk*

About me

- Call me Baris
  - *Or rather, 1st approximation: Barish*
  - *Exact solution: https://forvo.com/word/bar%C4%B1%C5%9F/*
- Assistant Professor
  - *Joined Michigan in Fall'17*
  - *PhD from EPFL*
  - *Previously, researcher at Microsoft Research*
- Interests: system reliability, security, performance
  - *Employ a mix of methods from Operating Systems, Programming Languages, Software Engineering, Computer Architecture*

Root Cause Diagnosis

Detection

Classification

[OSDI'18]
[SOSP'17]
[SOSP'15]
[HotOS'15]
[HotOS'13]

[SOSP'13]
[Wodet'14]
[HotDep'12]

[ASPLOS'12]
[TOPLAS'15]

Efficient Runtime Monitoring

[Usenix ATC'14]

BBC NEWS

Technology

FORESHADOW

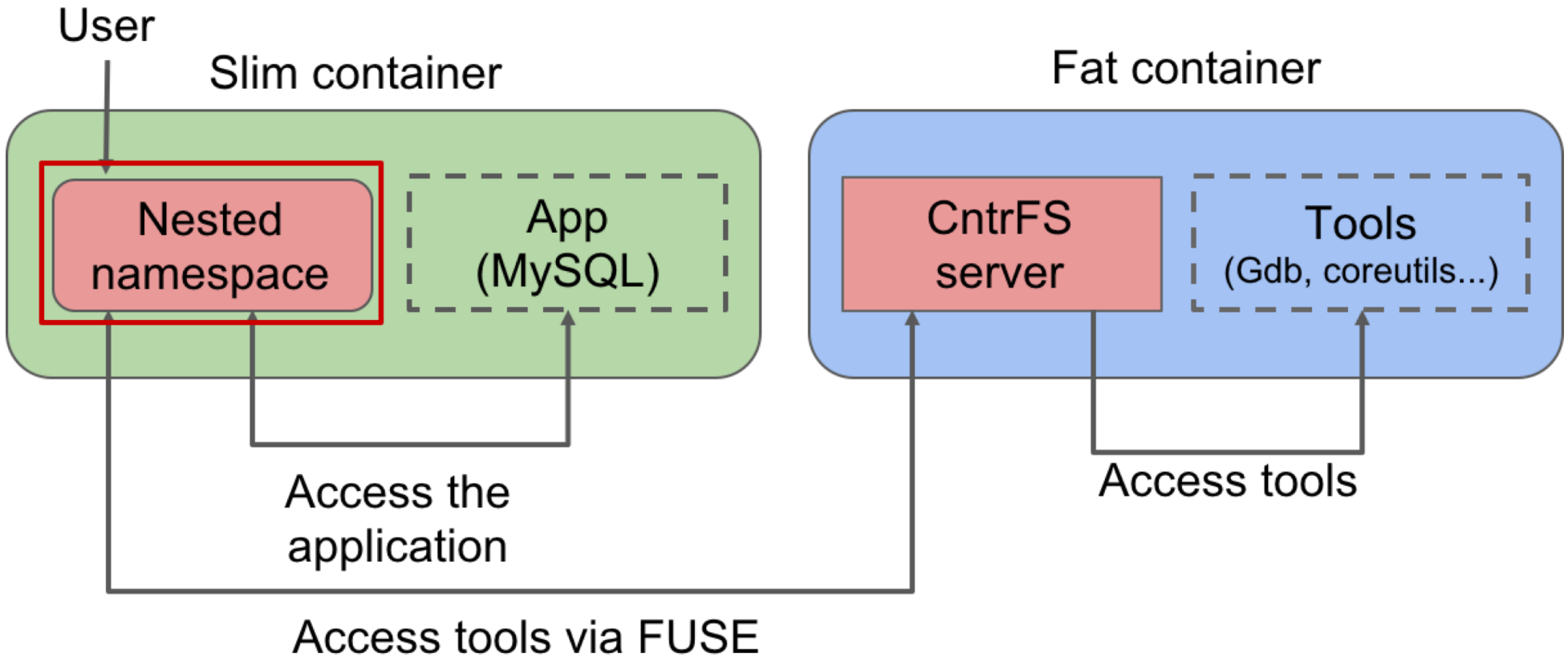
Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution

'Foreshadow' attack affects Intel chips

Beyond Spectre: Foreshadow, a new Intel security problem

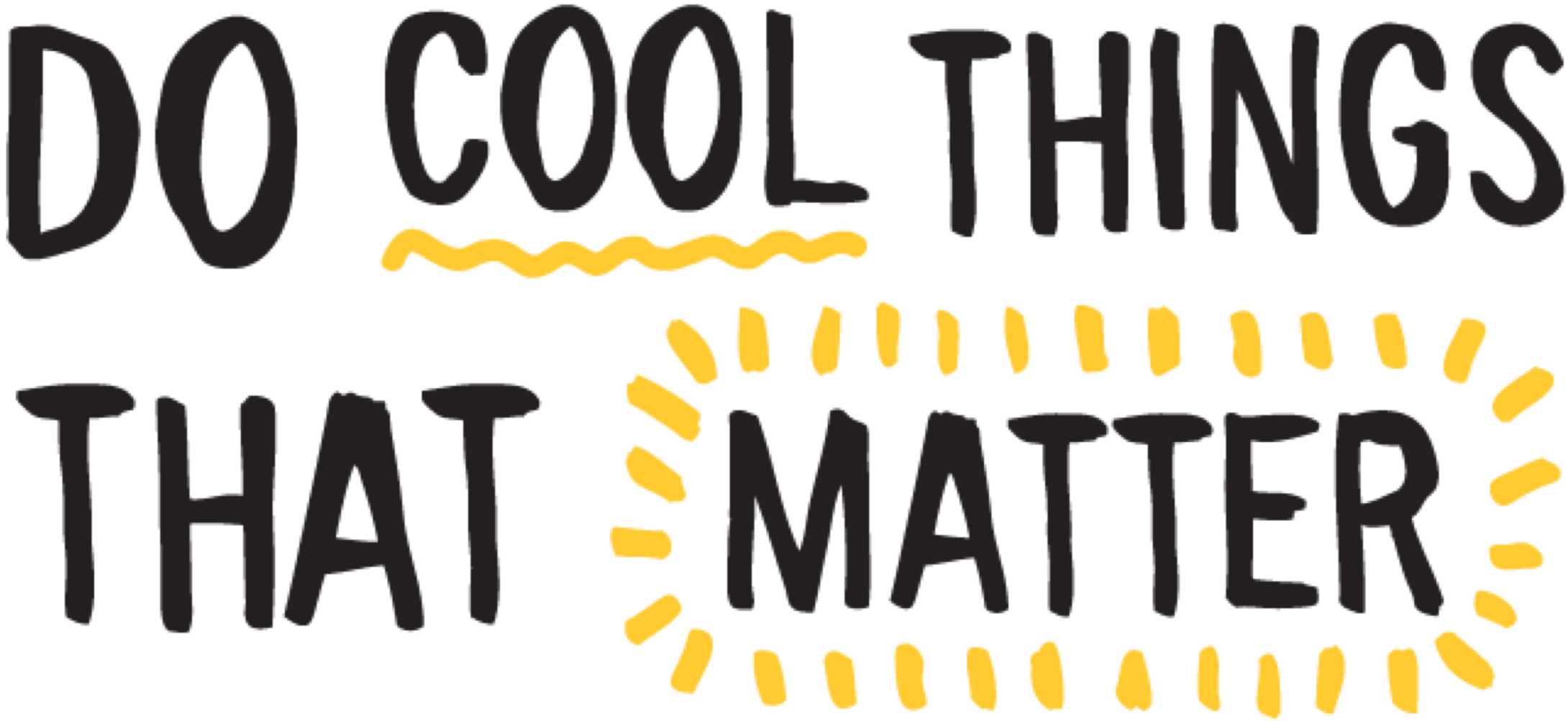
ForeshadowAttack.com

Cntr- A new approach to containers



Agenda

- **Introduction to (Systems) Research**
- Administrivia
- Project details
- Project ideas
- Scientific Method in Systems Research



[1] https://www.google.com/intl/es/about/careers/lifeatgoogle/do-cool-things-that-matter.html

Systems Research

- Work in OS, file systems, databases, networking, language run-times, system security, cloud computing, distributed systems…
- Not a “hard” science
  - No ground truth to be discovered
  - Things can be “sort of” right
- Key skill: critical thinking

What is a Systems Researcher?

- Alto personal computer (1972)
- Ethernet (1973)
- Mesa programming language (1975)
- Bravo text editor (1973)
- Interpress language (1980)
- Fast RPC (1987)
- Autonet (1987)
- Virtual book (1994)

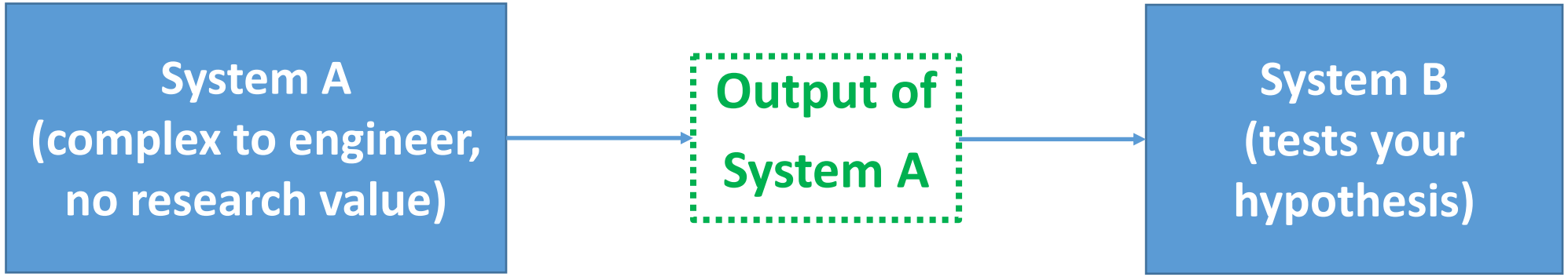
Butler Lampson

[1] http://amturing.acm.org/award_winners/lampson_1142421.cfm

Systems Challenges

- Greatest challenge: Complexity
  - *Complex preconditions and postconditions*
- Greatest problem: poorly understood connections
  - *Unexplained interface -> loss of predictability*
  - *Need predictability in order to achieve scalable performance, reliable operation, etc.*

Abstraction and Composition



Some Systems Solutions

- Caching
  - *Improve performance by adding another layer*
- Transactions
  - *Building blocks for robust systems*
- Virtualization
  - *Consolidation and isolation*

Systems Approach

- Get an idea
- Build prototype
- Measure & observe
- Adjust prototype and repeat the previous step

Apply principles of system construction

- *Modularity, composition, hierarchy, layering, abstraction, caching, transactions, end-to-end principle, etc.*

EECS 582 Goals

- Learn how to do (systems) research
- Understand the systems research process
  - *By studying successes and failures*
- Learn about classical and modern systems
  - *Unix*
  - *MapReduce*
  - *Xen*
  - *TensorFlow*
  - …

Agenda

- Introduction to (Systems) Research
- **Administrivia**
- Project details
- Project ideas
- Scientific Method in Systems Research

Prerequisites

- Formal prerequisites
  - *EECS 482 (Introduction to Operating Systems) or equivalent*
    - Basics of OS organization, threads, memory management, file systems, scheduling, networking, etc.
- Informal prerequisites
  - *Excellent programming skills*
    - You will build a substantial working prototype for the course project
    - Experience as a real-world developer (job in industry, internships, etc.)

Meetings and Schedule

- 1303 EECS M/W 3:00-4:30 PM
  - *Next week, we will decide how to structure the breaks together*
  - *Fridays: readings and project work*
    - Typically no meetings, except potential makeups
- Slides will be posted after the lecture
- Lecture starts at the hour (not Michigan time, i.e., 10 past the hour)
  - *Presentation of paper(s), (short break), discussions led-by presenters*
- Course webpage: http://web.eecs.umich.edu/~barisk/teaching
  - *Pay attention to the online announcements and the schedule*

Enrollment and Waitlist Status

- 43 enrolled, 13 waiting permissions
  - *Participation is key!*
- Course audience
  - *Graduate students interested in systems research*
  - *Typically not advisable to undergrads (definitely not if they haven't taken OS)*
  - *If in doubt, check with the instructor*
- **If you are not planning to take the class, please drop ASAP**

Grading

Paper Reviews	15%
Presentations	10%
Participation (you can skip 2 lectures for legitimate reasons)	10%
Midterm Exam (format TBD)	15%
Project (original contribution, suitable for publication)	50%

EECS 582 Topics

- Classics
- Kernel Design
- Naming
- Locality
- Performance Analysis
- Virtualization
- Concurrency
- Atomicity & Consistency
- Fault tolerance
- Verification and Testing
- Security
- Privacy, Censorship, Surveillance
- Datacenter Systems
- Storage
- New OS Designs
- Heterogeneous Systems

Reading Material

- ~60 papers, videos, and articles
  - *Many seminal, award winning papers*
- 19 paper reviews to write (Marked with (R) on the schedule)
- Read/watch all the required material before the lecture
  - *Participate, ask questions!*
- First decide whether you will take the class and then fill in https://goo.gl/forms/3QvYHQjIvwTWczFs1 to sign up for paper presentations
  - *Sign in with your UMich alias*

Paper Review Format

- Check out the reading list of Friday
- Summary format: https://gist.github.com/kasikci/49e7107dfdee281d6f6450b132555550
  - *What is the problem? [Good papers generally solve a single problem]*
  - *Summary [Up to 3 sentences]*
  - *Key insights [Up to 2 insights]*
    - "Aha!" moments, key observations leading to the solution
  - *Notable Design Details/Strengths [Up to 2 design details/strengths]*
  - *Limitations/Weaknesses [Up to 2 limitations/weaknesses]*
  - *Summary of Key Results [Up to 3 results]*
  - *Open questions [Where to go from here?]*

Paper Review Submissions

- Submit at: http://eecs582.eecs.umich.edu/papers/
  - *Create an account if you haven't*
- Deadline is midnight before the lecture
- You can miss at most 4 review submissions without any penalty
  - *Each miss beyond that will result in 25% decrease in grade for this portion of the course*
  - *Meaning, missing eight or more will result in 0% for the "Paper Reviews" segment of your grade*
- No extensions

Paper Presentations

- Each student must present at least one paper
  - *You will likely present another paper as part of a group*
  - *10% of the grade*
- Watch How to give a great research talk? (check the reading list too)
  - *Prepare early, practice a lot!*
- Email your slides to the instructor 24 hours before the class
- Use this PowerPoint template
- 20 minutes (**sharp**) of presentation, followed by a discussion led by the presenter (I advise you to prepare additional slides for this)
  - *Focus on strengths, weaknesses, and open questions from the paper review*
  - *Supplement discussion with auxiliary materials (papers, blog posts, etc.)*

A Great Research Presentation

- Motivate and describe the **problem**
  - *Why is this a scientifically interesting problem?*
  - *Who hurts if this problem is not solved?*
- Focus on **key insights and design details** behind the solution
  - *Skip low-level implementation details*
- Summarize the key **results**
  - *Don't just put up graphs, explain what do we learn from the key results*

Participation Policy

- Try to attend all lectures
  - *You can miss at most two lectures with legitimate reasons*
- Don’t fall behind in your reading material
  - You must read the papers before coming to class
- Ask questions
  - *To speakers during/after the talk*
- Engage in post-talk discussions
  - *Point out strengths and weaknesses that you identified*
  - *Suggest further future work*

Midterm Exam

- Format TBD
- Questions about system design
  - *Focus on design decisions trade-offs, and justifications*

Agenda

- Introduction to (Systems) Research
- Administrivia
- **Project details**
- Project ideas
- Scientific Method in Systems Research

Project

- The most substantial part of the course (50% of the grade)
  - *Done in groups of 2-3 students*
- Pick an interesting problem that you are interested in
  - *I can provide ideas and help refine your projects*
  - *Start thinking about your project today!*
  - *Start working on it tomorrow* ☺
- Research-related work
  - *What has been done, what remains to be done?*
- Devise a solution
- Build a substantial prototype
  - *Run experiments, measure, and compare to state of the art*

Project Scope

- Literature surveys cannot be projects
  - *Surveys are a prerequisite to coming up with research projects*
  - *Every project paper must include background survey (i.e., related work)*
- Projects address weaknesses and open questions identified in papers
- Ideally, you already have research interests
  - *You can frame a project fitting your interests and the subjects we cover*
  - *Consult with me, your advisor, and your group when framing the project*

Project Process

1. Find a problem that matters
2. Read background information to get a sense of the state of the art
   - *Distribute the work among your group members for efficiency*
3. Go back to 1 until you define and refine the problem
4. Formulate/update your hypothesis
5. Go back to 4 until you are satisfied
6. Present your findings in a poster and a paper

Strong Inference
(see the schedule for other readings)

Project Milestones

Date	Milestone	Details
ASAP	Form group	Find 2-3 like-minded students
09/21/17	Draft proposal	Send your proposal to the instructor by email
10/03/17	Finalize proposal	After regular back-and-forth discussions with the instructor
11/5/17 11/7/17	Mid-semester presentations	Define and motivate a problem, survey related work, and form initial hypothesis and idea
12/12/17	Poster presentations	Present your findings in a poster session
12/19/17	Research paper	Submit a project paper similar to the ones you read

Draft Proposal

- Max 2 pages with references
- **Ideally** includes:
  - *What is the problem?*
  - *Why does it matter?*
  - *How do you intend to solve it?*
  - *How will you evaluate your solution?*
- After submitting the draft, schedule a 15 minute meeting with the instructor to discuss the draft
  - *Refine the draft by iteratively communicating with the instructor*

Final Proposal

- Max 2 pages with references
- **Must** include:
  - *What is the problem?*
  - *Why does it matter?*
  - *How do you intend to solve it?*
  - *How will you evaluate your solution?*
- To be approved by the instructor and agreed upon by you
  - *Forms the basis of expectation for your grade*

Mid-Semester Presentation

- Short in-class presentations to track your progress
  - *We will determine the duration based on the number of groups*
- Present your early work and get feedback
- Presentation format
  - *What is the problem?*
  - *Why does the problem matter?*
  - *What is the closest related work?*
  - *What is your hypothesis?*
  - *What is your progress so far?*
  - *How will you evaluate your solution?*
  - *What are the challenges you are currently facing?*

Final Poster and Paper

- Posters are a good way to interact with others
  - *Most top conferences will have a poster session*
  - *It is a crucial research skill to be able to talk about your work*
- Research paper
  - *Main component of the course*
  - *Similar to the papers you have read*
  - *~8 pages*
- Check the reading list for guidelines on poster and paper preparation

Rough Paper Outline - 1

- Abstract (2-3 paragraphs)
- Introduction (~1 page)
  - *Define and motivate the problem with examples, briefly talk about challenges*
  - *Explain your core high-level idea*
  - *Give an overview of your solution and summarize key results*
- Design (~3 pages)
  - *Explain the core technical details of your project*
- Implementation(~0.5 page)
  - *Explain subtle implementation details*

Rough Paper Outline - 2

- Evaluation (~1.5 page)
  - *Convince the reader that your system works and explain when it fails*
- Related work (0.5-1 page)
  - *Know your competition, and show how you differ from it*
- Discussion (0.5 page)
  - *Discuss your limitations and possible workarounds*
- Conclusion (2-3 paragraphs)
  - *Summarize the paper and briefly point out the future work*

Action Items for the Next Lecture

- Please fill out: https://goo.gl/forms/3QvYHQjIvwTWczFs1
  - *Try to decide if you will drop the course **before** filling the form*
  - *Includes your presentation preferences*
- Create an account: http://eecs582.eecs.umich.edu/papers/
  - *We will use HotCrp to manage paper reviews*
- Read/watch all the required material
  - *Read/watch today's material as well if you haven't already*

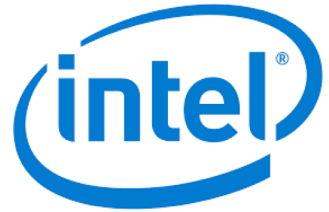
Agenda

- Introduction to (Systems) Research
- Administrivia
- Project details
- **Project ideas**
- Scientific Method in Systems Research

Project Ideas

- Non-exhaustive
  - *Topics I am working on/interested in*
- I encourage you to define your own project topic
  - *Systems research is broad*
  - *Projects can be formulated to fit your research area/interests*
  - *Open to collaboration with your advisors*

Performance Analysis and Optimization



- AutoFDO (one of the required readings)
  - *Automatically optimizing programs by using information gathered from hardware performance counters*
  - *Focuses on the text segment*
- Can we perform more holistic optimizations?
  - *Optimizing the data segment as well as the text segment*
- Can we perform lifelong program optimizations?
  - *Compile time as well as runtime*
- Can high-level design goals guide program structuring?
  - *E.g., data and instruction locality*

System Support for Heterogeneous Architectures

- Heterogeneity is increasing due to performance bottlenecks
  - *GPUs, accelerators, FPGAs*
- Managing heterogeneity at the system level is challenging
  - *How to parallelize an FPGA?*
- Programming heterogeneous systems is challenging
  - *What should the programming model be?*
  - *What are the proper interfaces?*
- Debugging heterogeneous systems is challenging
  - *Different memory models*

System Security

- Record/replay for debugging side channel attacks
- Secure computer and system architecture design
- Making hardware-based security more usable and useful
  - *Intel SGX, ARM TrustZone*
- Improving IoT Security
  - *Challenging because of limited resources and high degree of interconnectivity*
- Building a secure distributed shared memory system
- …

Debugging Cloud Systems

- Both correctness and performance debugging is important
  - *Automatic Troubleshooting in the Cloud*
  - *Request tracing in a Cloud environment*
- Analysis of data logs in Cloud systems
  - *Inferring state machines from logs*
  - *Studying normal behavior and identifying deviations using ML*

Hardware-Assisted Software Analysis

- Speeding up symbolic execution using branch tracing
  - *Useful for testing and. bug finding*
  - *Hardware support can improve the scalability of*
- Improving the accuracy of points-to analysis
  - *Can be used to improve dataflow analyses*
  - *Improve the accuracy and efficiency of Control Flow Integrity approaches*

Agenda

- Introduction to (Systems) Research
- Administrivia
- Project details
- Project ideas
- **Scientific Method in Systems Research**

  Feynman's Scientific Method Talk

Strong Inference

1. Devise hypotheses
2. Devise experiments that can refute hypotheses
   - *Some experiments are better than others in reducing the number of possibilities*
3. Refine & Iterate
   - *Reflection: what did we do wrong?*

[1] John R. Platt, Strong Inference, Science Oct. 1964

Smart Experiment Design

*"**If you do stupid experiments, and finish one a year, it can take 50 years**. **But if you stop doing experiments for a little while** and think how proteins can possibly be synthesized, there are only about 5 different ways, not 50! And it will take **only a few experiments** to distinguish these."*

*"It pays to have a top-notch **group debate** every experiment ahead of time"*

[1] John R. Platt, Strong Inference, Science Oct. 1964

Objectivity

*“When we make a **single** hypothesis, we **become attached** to it."*

> *“When **multiple hypotheses** become coupled to strong inference, the scientific search becomes an emotional powerhouse as well as an intellectual one.”*

Method Orientation vs Problem Orientation

Beware of the man of one method or one instrument, either experimental or theoretical. He tends to become method-oriented rather than problem-oriented. The method-oriented man is shackled: the problem-oriented man is at least reaching freely toward what is most important.

Fermi's Notebook Method

- Devote ½ hour everyday to analytical thinking
  - *Write out the logical tree*
  - *Write out the alternatives*
  - *Identify crucial experiments*