Seminar on Machine-Checked Correctness Proofs for Systems

Intro & Administrivia

Samuel Gruetter

Feb 17, 2025

About me

- Samuel Gruetter ("Sam")
- Background
  - PhD in Programming Languages & Formal Verification
  - Interactive Proof Assistants (Coq/Rocq)
  - End-to-end machine-checked proofs about behavior of software/hardware stacks
- Now: Postdoc in Prof Timothy (Mothy) Roscoe's group
  - Apply formal methods to Systems

Background

- Soft requirement: Class "Formal Methods and Functional Programming"
- In any case: Skim referenced papers, google/read up on background needed in papers

Topic of this seminar

- Today's software is full of bugs and vulnerabilities
- Can we *prove* that there are no bugs?
- But what if our proofs have bugs?
- *Machine-checked* proofs

How to prove software (or hardware) correctness

- First approach:
  - Design a simpler, "obviously correct" system ("the spec")
  - Prove that the implementation "behaves like" the spec
- Second approach:
  - Preconditions, postconditions, invariants

Why read papers?

- Learn about results & techniques that have not yet made it into textbooks
- Know what's already been done
  - to avoid duplicate research
  - to know what you can reuse in your own research (or product!)
- Gain deeper understanding of a problem

Why teach a seminar? (Hidden agenda)

- Recruit students for BSc/MSc/PhD projects with the Systems Group
- Force myself to read some papers in more detail that I wanted to read for a long time already

How to read a paper

Research papers vs. **Textbooks**

Slide credit: Ana Klimovic

Reading in several passes

- Don't read linearly, find your own approach

Notion: Trusted Code Base (TCB)

- Which code do you need to read and trust in order to know that the system behaves correctly?
- Example: Sandbox
- Example: Interactive proof assistant:
  - Proof-checking kernel
  - Untrusted proof-search heuristics

Questions to ask

- "We have proved X" = ???
- Trusted code base: What is explicitly/implicitly assumed to be correct?
- Generalizability
- Level of automation / Required user interaction
- Expressivity of language of assertions
- What cannot be expressed this assertion language?
- Benefits of restricting the expressivity
- What is the user's workflow?
- What if the tool cannot prove a property?
  Does it just say "failed"? Run forever? Error messages?
- How would you do it? What would you do differently from the paper you just read?

"C’est quoi l’arnaque?" *

- "What's the scam?"
- Reading a paper:
  - Impressive title & abstract
  - Hold on, that's too good to be true!
  - What's the scam?
  - Ahaa, these are the deliberately chosen limitations
- There are good scams and bad scams
- Often, choosing a reasonable set of limitations is a valuable contribution!

* Wisdom by Thomas Bourgeat

Example "Scam"

- Claim: We can solve SAT efficiently
- Wait, but: This problem is NP-complete (we don’t know how to solve faster than in than exponential time)
- “Scam”: Still exponential in the worst case, but, empirically, fast for many practically relevant instances

Example "Scam"

- Claim: We can automatically tell if an assembly program satisfies a spec
- Wait, but: Contradicts Rice's theorem: All non-trivial semantic properties of programs are undecidable
- Scam: Only bounded loops, and to make it even simpler, no I/O or nondeterminism, no system calls, ...

Intro Round

- Nametags

Organization

- Each student co-presents 2 papers
- Each paper is presented by a team of 2 students
- 45 min per paper. Procedure:
  1) Explain the paper: split between the two students
  2) Student 1: Sell the paper:
     - How does the paper advance the field?
     - What does the paper do well?
  3) Student 2: Critique the paper:
     - What are its limitations?
     - How could it be improved?
  4) Discussion, moderated by the two students
- 1)-3) should take ca 25 min, 4) ca 20 min

Preparation of Presentation

- 1 week before presentation: email to samuel.gruetter@inf.ethz.ch:
  - summary
  - strengths
  - limitations
  - optional questions
- By Monday 1pm, get feedback in-person before or after session

Grading/Goals

- My goal: Attending this seminar should be a “good deal” for you:
  - Learn about exciting topics
  - Improve paper reading, presenting, discussion skills
  - Get 2 ECTS & good grade in exchange for a manageable time investment
- Grade criteria:
  - Most important skill: Can you articulate what you do not understand?
  - Presentation: Explained well? Identified strengths & limitations?
  - Participation in discussions

Attention

- If not interested: Don’t miss the deadline for deregistering. (The VVZ says: “*The deadline for deregistering expires at the end of the second week of the semester. Students who are still registered after that date, but do not attend the seminar, will officially fail the seminar.*”)

Some advice about slides…

Slide credit: Ana Klimovic

What's wrong with this slide?

Strengths

- The paper addresses a very important problem: how to efficiently process training data before it is fed to a machine learning model for training. Prior work has focused on accelerating training computations, however data preprocessing is also critical for end-to-end training performance and cost.
- The system is designed to be easy-to-use, requiring minimal changes to applications. Users also do not need to worry about resource provisioning; the system automatically manages resources.
- The paper is very well-written. The use of examples is very effective for explaining how the system works.
- The paper includes an interesting analysis of production workloads at a large-scale cloud provider. The authors discuss implications of their findings for the design of their system and future work, such as processing data closer to storage or distributing data processing to multiple workers.

Slide credit: Ana Klimovic

How to improve this slide?

Strengths

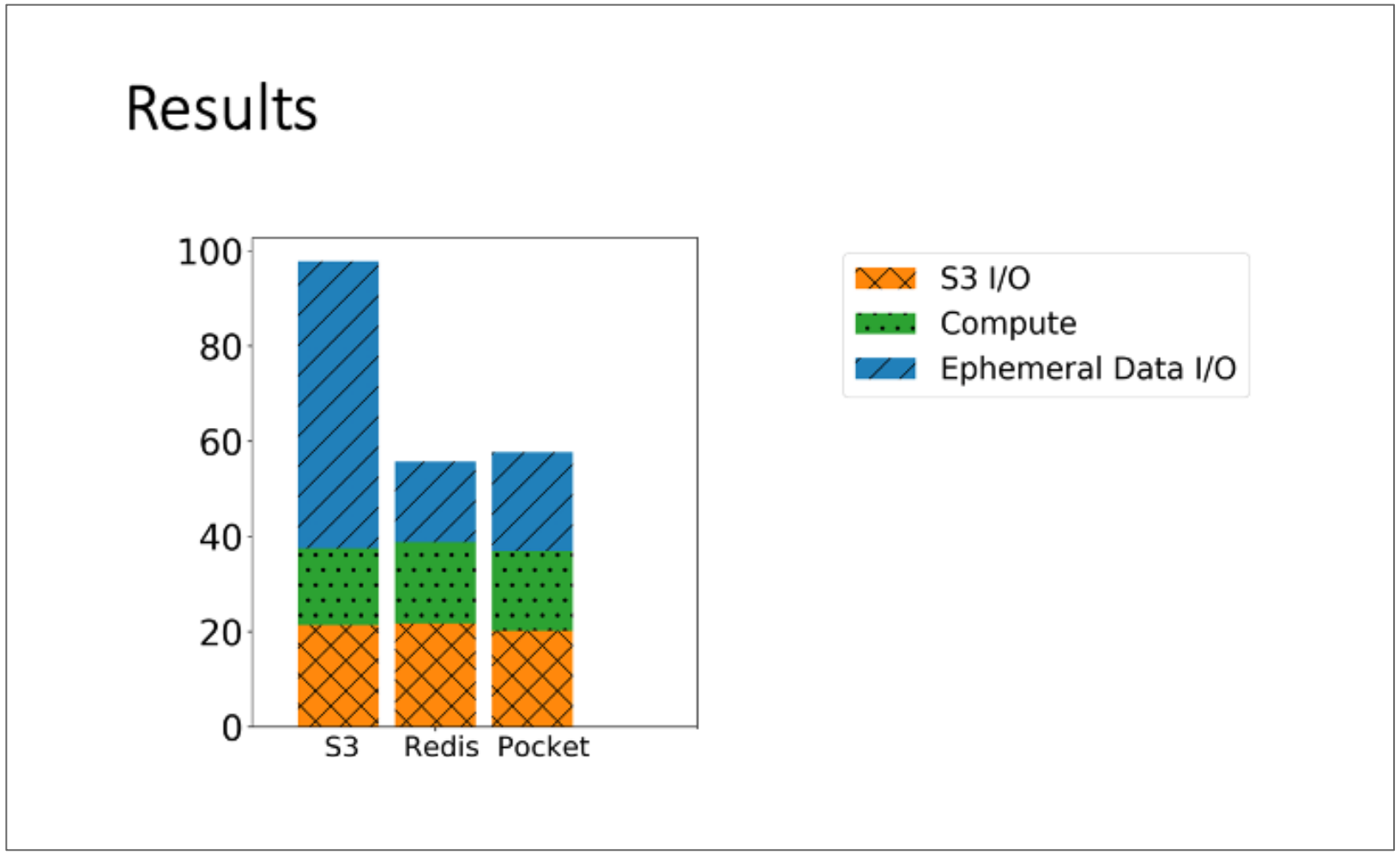
- The paper addresses a very important problem: how to efficiently process training data before it is fed to a machine learning model for training. Prior work has focused on accelerating training computations, however data preprocessing is also critical for end-to-end training performance and cost.
- The system is designed to be easy-to-use, requiring minimal changes to applications. Users also do not need to worry about resource provisioning; the system automatically manages resources.
- The paper is very well-written. The use of examples is very effective for explaining how the system works.
- The paper includes an interesting analysis of production workloads at a large-scale cloud provider. The authors discuss implications of their findings for the design of their system and future work, such as processing data closer to storage or distributing data processing to multiple workers.

Reduce the volume of text.
You don't need to include everything that you will say!
You also don't need full sentences.

Show diagrams to illustrate points rather than only relying on text.

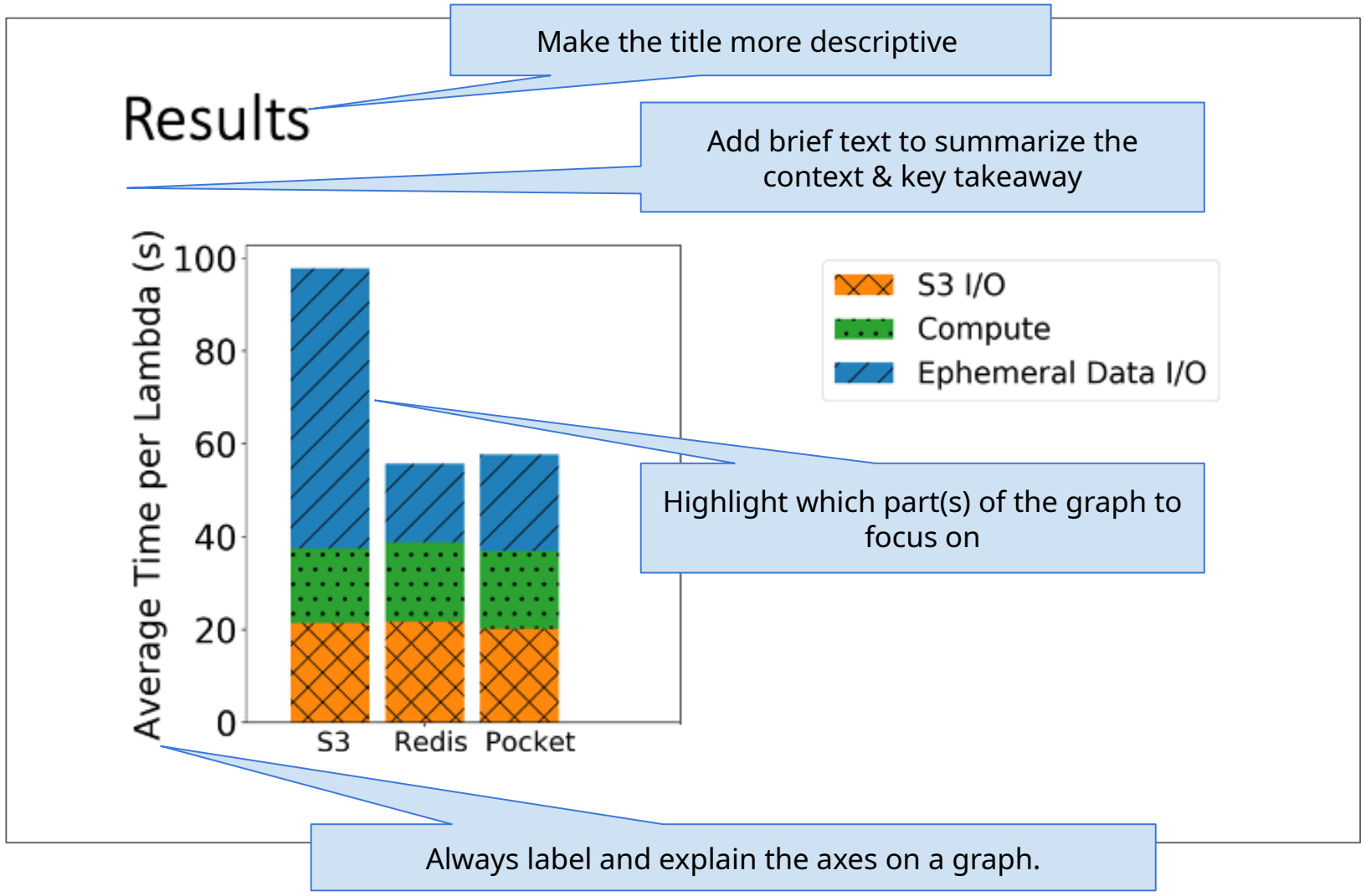
Slide credit: Ana Klimovic

What’s wrong with this slide?



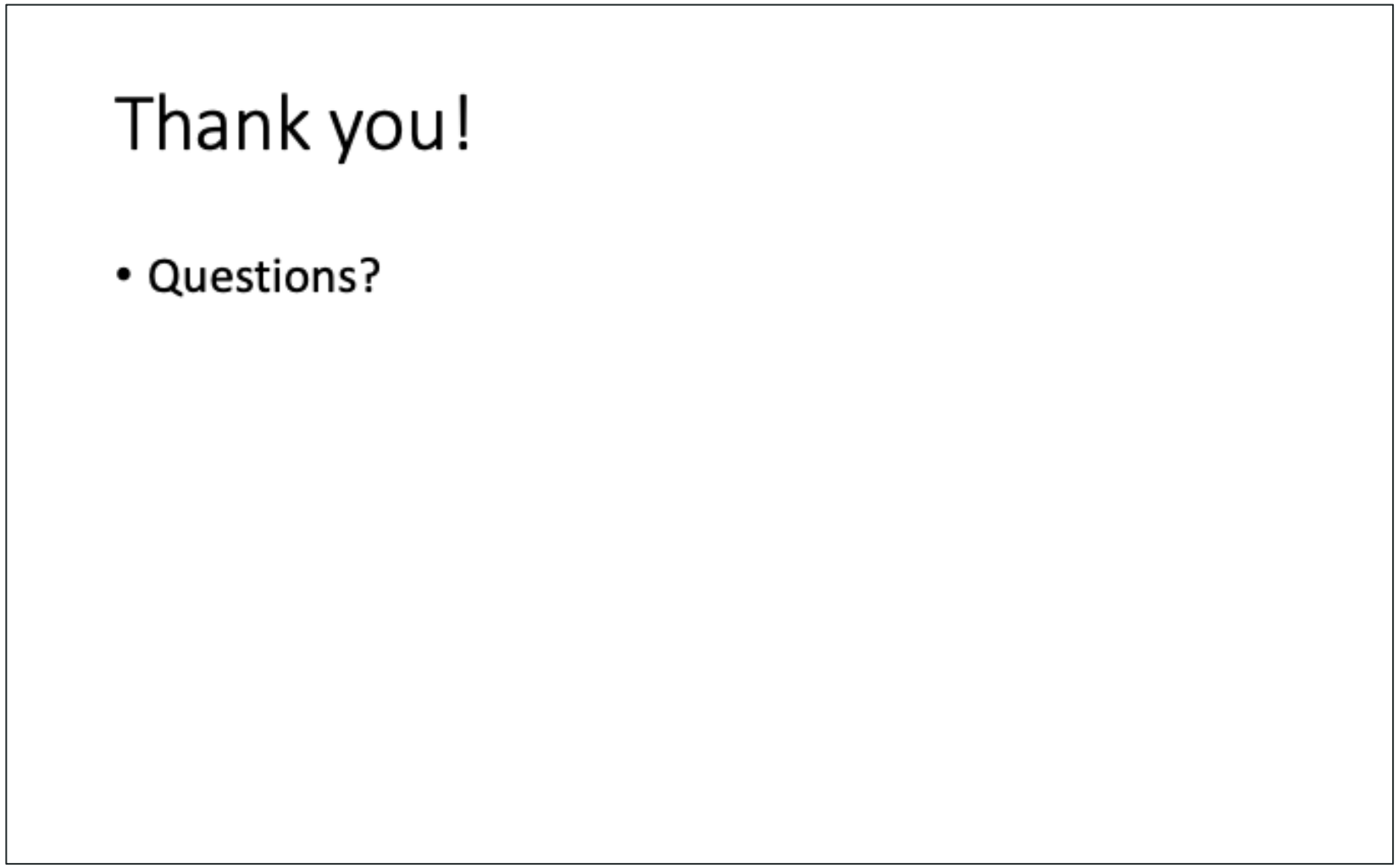
Slide credit: Ana Klimovic

How to improve this slide?



Slide credit: Ana Klimovic

What's wrong with this slide?



Slide credit: Ana Klimovic

How to improve this slide?

Slide credit: Ana Klimovic

Conclusion/Your TODOs

- Present 2 papers, in teams of 2
- Sell 1x, Critique 1x
- 1 week before presentation: email to samuel.gruetter@inf.ethz.ch summary/strengths/limitations/optional questions by 1pm, get feedback in-person
- Show up to each session & participate in discussion
- Quickly read papers in advance, prepare at least 1 question
- Most important learning goal:
  Be able to articulate what you do not understand