What is systems research?

a not-very-serious introduction

Tej Chajed

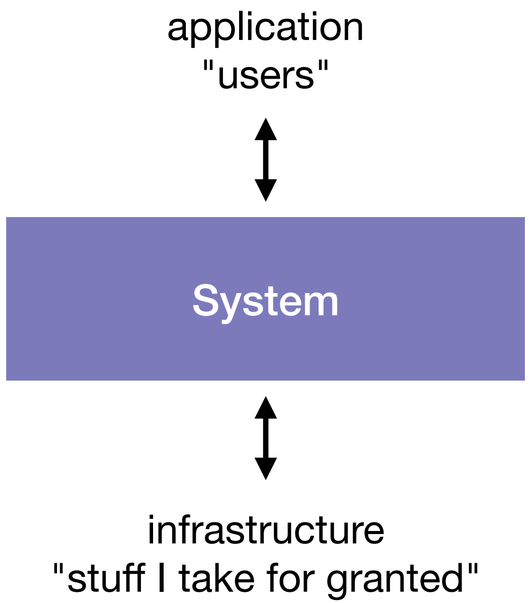
Methodology for this talk

Spend many years conducting systems research

Spend an hour going through conference programs

Spend twenty minutes making slides

Software "stack"



One person's system is another's infrastructure (and someone else's application)

Systems makes computers useful

Two pertinent questions:

- Is this system useful?
- Does it meet performance requirements?

Theme 1: new <system> for <domain>

Systems

Databases, operating systems,
programming languages,
schedulers, networking

Framing borrowed from Ilya Sergey's "How to Bootstrap a Research Project"

Theme 1: new <system> for <domain>

Systems

Databases, operating systems, programming languages, schedulers, networking

Application domains

Mobile code, scientific computing, web applications, order processing

Framing borrowed from Ilya Sergey's "How to Bootstrap a Research Project"

Theme 1: new <system> for <domain>

Systems

Databases, operating systems, programming languages, schedulers, networking

Application domains

Mobile code, scientific computing, web applications, order processing

Just kidding, it's all machine learning

Framing borrowed from Ilya Sergey's "How to Bootstrap a Research Project"

Theme 2: systems have many competing requirements

Throughput, tail latency, scalability, efficiency

Features, correctness, reliability, security

Theme 2: systems have many competing requirements

"performance"	Throughput, tail latency, scalability, efficiency
"other stuff"	Features, correctness, reliability, security

Theme 3: Systems researchers build systems

Every paper at OSDI and SOSP involves writing and running code

What is programming languages research?

still not very serious

Theme : <technique> applied to <domain>

Tools and techniques

Static analysis, semantics, abstract interpretation, program logic, type system, domain-specific languages, synthesis

Theme : <technique> applied to <domain>

Tools and techniques

Static analysis, semantics, abstract interpretation, program logic, type system, domain-specific languages, synthesis

Application domains

Smart contracts, web applications, mobile code, distributed systems, streaming systems, probabilistic programming

Theme : <technique> applied to <domain>

Tools and techniques

Static analysis, semantics, abstract interpretation, program logic, type system, domain-specific languages, synthesis

Application domains

Smart contracts, web applications, mobile code, distributed systems, streaming systems, probabilistic programming

Just kidding, it's all machine learning

Theme 2: emphasis on reasoning about programs

Slightly different questions asked:

- Applicability: can we reason about many programs?
- Theory: is our reasoning correct? is it elegant?

Theme 2: emphasis on reasoning about programs

Slightly different questions asked:

- Applicability: can we reason about many programs?
- Theory: is our reasoning correct? is it elegant?

Sometimes people design languages, too

Theme 3: PL researchers build tools

Not always, but PL research often involves implementation

Not always, but PL research also involves proving things

Systems research is about coming up with new ways of organizing software to make infrastructure useful

computers are just carefully organized sand, after all

Conferences

Systems:

- core systems: OSDI / SOSP (every other year), EuroSys
- security: eg, USENIX Security, IEEE S&P ("Oakland")
- architecture: eg, ISCA

PL:

- core PL: POPL, PLDI, ICFP, SPLASH (formerly OOPSLA)
- verification: eg, CAV

Every systems paper outline

most important section → 1. Introduction: (what is the problem? what are the contributions?)

2nd-most important section →

Every systems paper outline

most important section →

1. Introduction: (what is the problem? what are the contributions?)

2. Related work (appease reviewers)

2nd-most important section →

Every systems paper outline

most important section →

1. Introduction: (what is the problem? what are the contributions?)

2. Related work (appease reviewers)

3. Approach (high-level, broad audience)

2nd-most important section →

Every systems paper outline

most important section →

1. Introduction: (what is the problem? what are the contributions?)

2. Related work (appease reviewers)

3. Approach (high-level, broad audience)

4. System design (everyone skips this)

2nd-most important section →

Every systems paper outline

most important section →

1. Introduction: (what is the problem? what are the contributions?)

2. Related work (appease reviewers)

3. Approach (high-level, broad audience)

4. System design (everyone skips this)

5. Implementation (we wrote 10,000 lines of C code...)

2nd-most important section →

Every systems paper outline

most important section →

1. Introduction: (what is the problem? what are the contributions?)

2. Related work (appease reviewers)

3. Approach (high-level, broad audience)

4. System design (everyone skips this)

5. Implementation (we wrote 10,000 lines of C code...)

2nd-most important section →

6. Evaluation (everyone starts here)