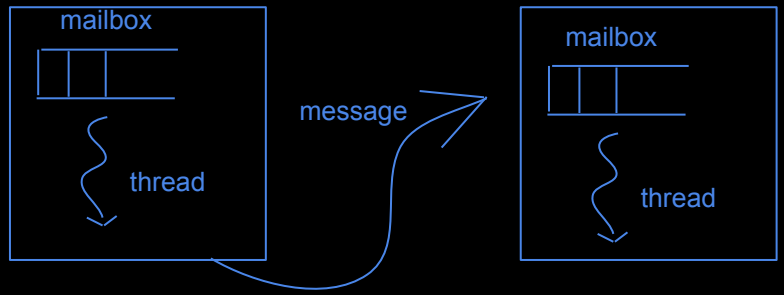
Actor Programming with Static Progress and Safety Guarantees

Minas Charalambides
Open Systems Laboratory
University of Illinois at Urbana-Champaign



Actors

- Encapsulated, autonomous, concurrent active objects.
- Communicate via asynchronous message passing.
  - Full encapsulation, so no shared memory.

Issues

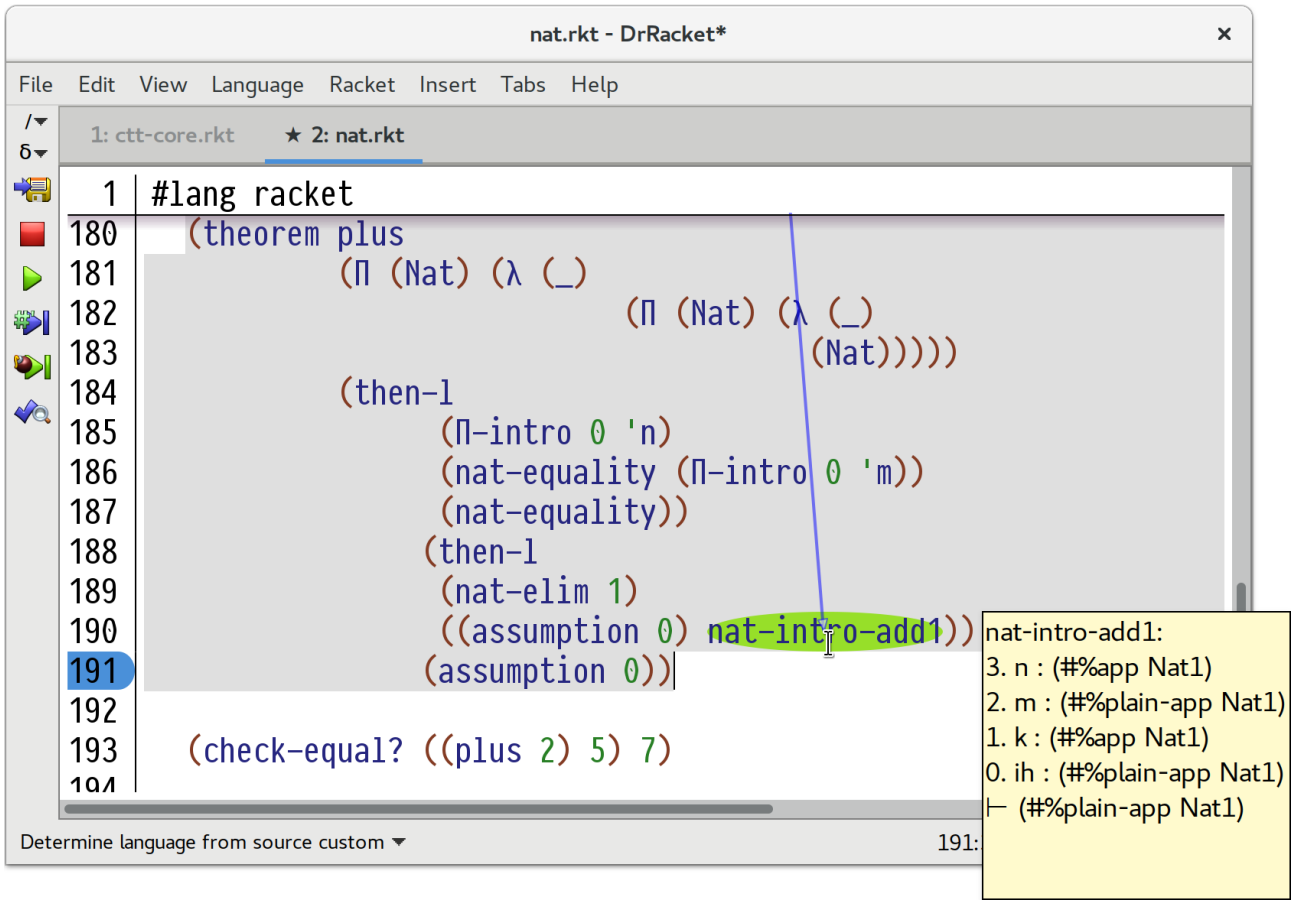
- Deadlocks, livelocks, etc.
- Communication safety.
  - Avoid undesired situations.
  - e.g. full buffer vs empty buffer

Proposed solution

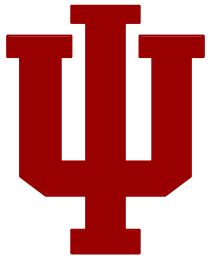
- Uses types.
- Tokens for obligations and requirements.

Proof Assistants as Macros

David Christiansen Indiana University



- Accurate bindings
- Proofs about macros
- Tool integration



PURDUE
UNIVERSITY

Fauxquet: Parquet in Scala

James Decker and Tiark Rompf

We look at Apache Parquet, a compressed, column store, file format used in the Hadoop ecosystem in big data processing. We reduce some inefficiencies in this project (including a speedup of up to 25x), as well as move the format to Scala. Our future work includes taking a generative, multi-staged approach to this format to further increase our efficiency gains as part of the Flare ecosystem at Purdue University.

James Decker wishes to express his regret that he is unable to attend today due to a family emergency. Come swing by the poster anyway for more information!

The Effect of Instruction Padding on SFI Overhead

Navid Emamdoost

Software-based Fault Isolation

Google Native Client

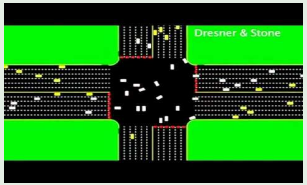
- Reducing instruction padding
- Updating verifier

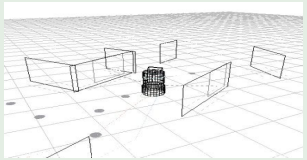
Coq proof

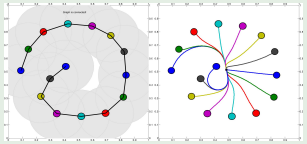
PROFILE

University of Minnesota

PCCL: Physical Coordination and Control Language







In seconds, with guarantees

Heterogeneous platform

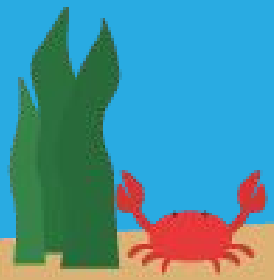
Ritwika Ghosh
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign.

ORC²A Proof Assistant

A new proof assistant for pedagogical use in introductory computer science courses

Grinnell College, Iowa

Jerry Chen, Medha Gopalaswamy,
Sooji Son, Peter-Michael Osera



Finding Races Due to Asynchrony in Mobile Applications

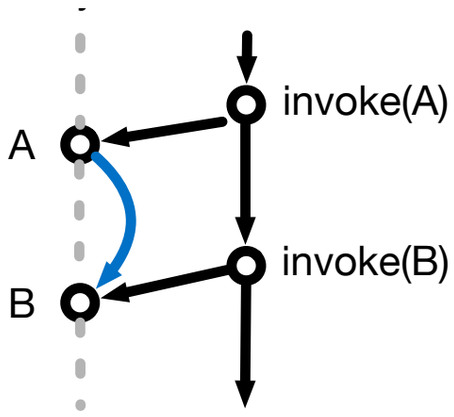
COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

Chun-Hung Hsiao
Satish Narayanasamy

intel
Cristiano Pereira
Gilles Pokam

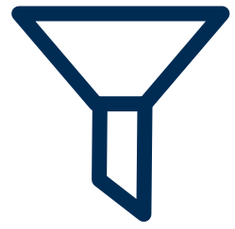
- Important class: mobile, web, autonomous vehicles, …
- New class of concurrency errors to asynchrony

1. Causality Inference



2. Race Detection

3. Commutativity Filter



- Found 147 data races in 20 applications. [PLDI'14, ASPLOS'17]

Seonmo Kim

- Ph.D student at University of Minnesota
- Title: Bit-Vector Model Counting using Statistical Estimation
  - SearchMC: Approximate model counter for CNF and SMT formulas
  - Quantitative information flow

  - https://github.com/seonmokim/SearchMC

DCatch Poster Introduction

Haopeng Liu from Prof. Shan Lu's group at University of Chicago

1. DCbugs

   -- Distributed timing bugs widely existing in cloud systems.

   -- Unique scalability, coverage, and accuracy challenges to bug detection.

2. DCatch

   -- A new Happens-Before model for real-world cloud systems.

   -- An effective tool to detect DCbugs during correct runs.

This work will appear at ASPLOS 2017

unit-tested

compiler plug-in

Affordable 2nd-class values for fun and (co-)effect

Scala

proofs in Coq

Leo Osvald, Grégory Essertel, Xilun Wu, Lilliam I. González Alayón, and Tiark Rompf

PurPL

Object-Oriented

re-intro from Algol/Pascal

stack-based lifetimes

Atul Sandur

2nd year Computer Science PhD student, Advisor: Prof. Gul Agha
University of Illinois at Urbana-Champaign

Poster: Programming Large Scale IoT Applications

Keywords: *adaptive control, simulation, distributed monitoring, probabilistic programming, model checking*

Finding Semantically-Equivalent Binary Code By Synthesizing Adaptors

Vaibhav Sharma

```
int musl_isalpha(int c) { return ((unsigned)c|32)-'a' < 26; }

int glibc_isalpha(int c) { return table[c] & 1024; }
```

PURDUE UNIVERSITY

Gregory Essertel, PhD Student

I use generative programming to generate efficient and provable secure low-level code from high level language.

Ruby Tahboub, PhD Student

I apply PL and Compilers techniques to natively compile SQL queries and speed up query engines.

PURDUE
UNIVERSITY

Flare: Scaling-up Spark SQL with Native Compilation

In this work, we bridge the performance gap between Spark SQL and what can be achieved by best-of-breed query engines or hand-written low-level C code.

PURDUE UNIVERSITY

COMPUTER SCIENCE

Formal Proof of features in System $D_{<:}$

Fei Wang
Dr. Tiark Rompf

Key words:
PL calculus ($D_{<:}$)
Mechanical proof (Coq)
Strong Normalization
Grander Goals

We love Scala

Toward Fixed-Point Optimization in LLVM

Are we generating the most optimal code?

Nathan Wilson, KCG/UChicago
Hal Finkel, Argonne