Pavol Cerny

Bor-Yuh Evan Chang

Sriram Sankaranarayanan

Programming Languages and Verification at the University of Colorado Boulder

PLV

PLV research, right?

PLV research at CU has ***breadth***!

PLV research at CU has ***breadth***!

How do we assist reasoning about programs?

program analysis, developer tools

PLV research at CU has ***breadth***!

How do we assist reasoning about programs?

program analysis, developer tools

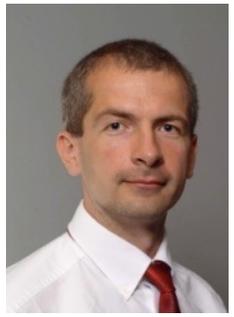
How do we get reliable, secure software?

verification, model checking

PLV research at CU has *breadth*!

How do we let computers code for us?

synthesis

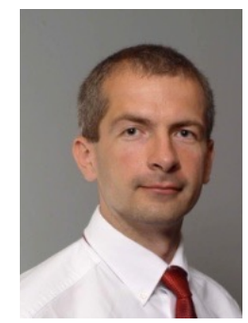



How do we assist reasoning about programs?

program analysis, developer tools


How do we get reliable, secure software?

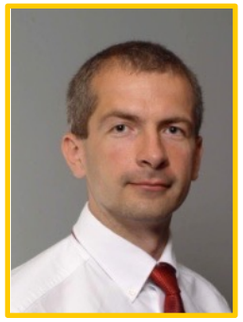
verification, model checking

PLV research at CU has *breadth*!

How do we let computers code for us?

synthesis




How do we assist reasoning about programs?

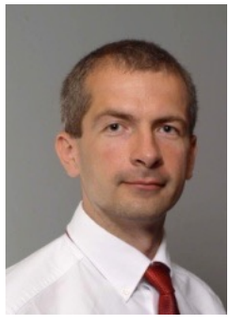
program analysis, developer tools


How do we get reliable, secure software?

verification, model checking


PLV researchers at CU *collaborate*!

synthesis

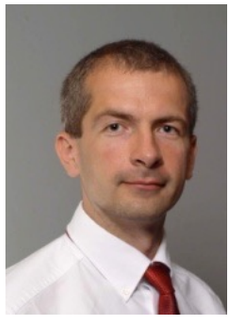


program analysis

verification

PLV researchers at CU *collaborate*!

synthesis



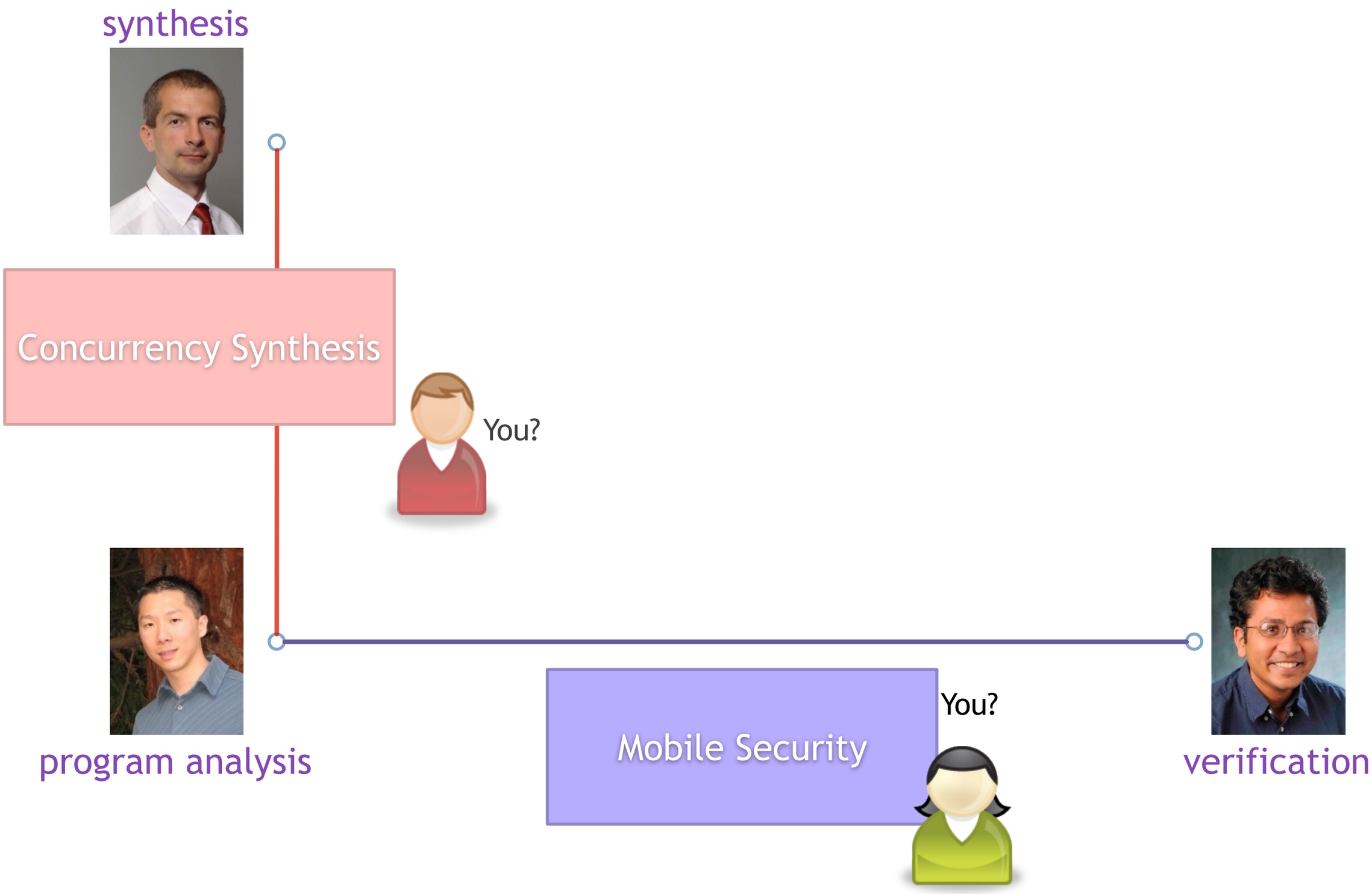
Concurrency Synthesis

You?

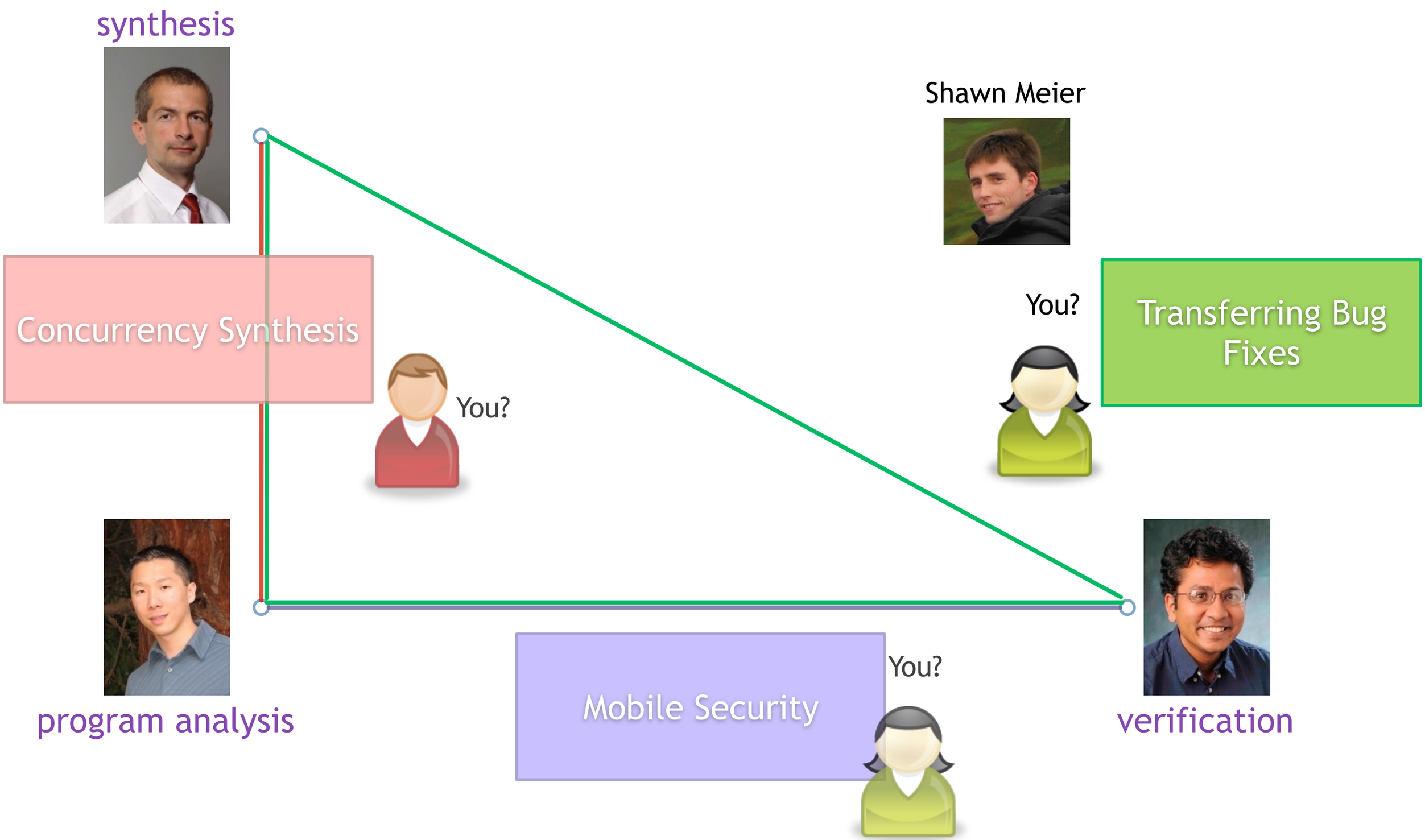
program analysis

verification

PLV researchers at CU *collaborate*!



PLV researchers at CU *collaborate*!




HOW'S THE DEBUGGING GOING?
JORGE CHAM © 2005
www.phdcomics.com

PLV is about compassion!

Do you Android?

Do you Android?

Do you Android?

Do you Android?

I’m not making this up …

I’m not making this up …

stackoverflow Questions Tags Tour Users

Android: Crash on rotation, horizontal to vertical

Crash is detected after rotating phone in Gmail Sync now view

phonegap ›

[important bug]cordova 1.9 crash on rotation android

5 posts by 2 authors +1

stackoverflow Questions Tags Tour Users

App crashes when rotating Samsung phone

androidterm

Android Terminal Emulator

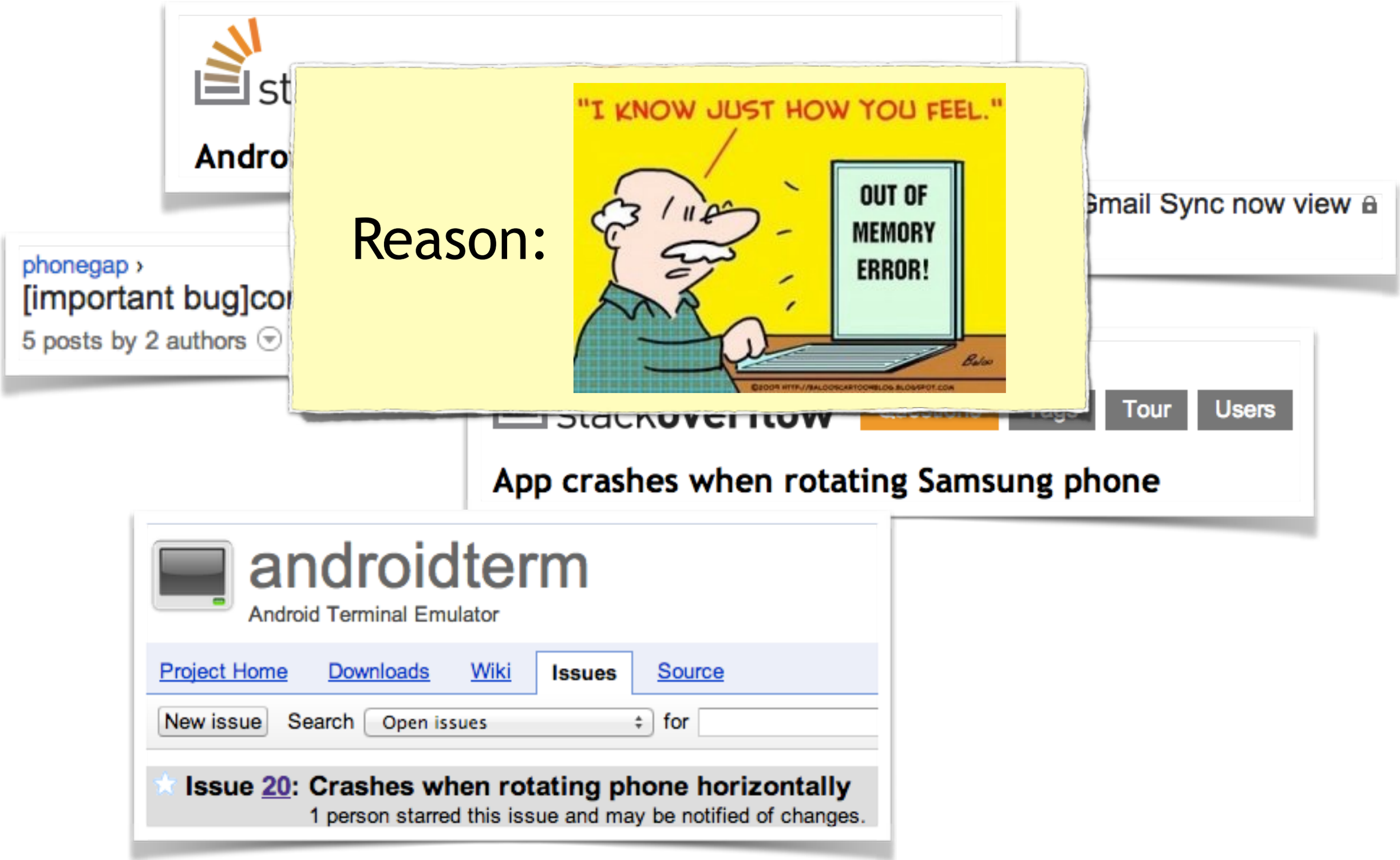
Project Home Downloads Wiki Issues Source

New issue Search Open issues for

Issue 20: Crashes when rotating phone horizontally

1 person starred this issue and may be notified of changes.

I’m not making this up …




CUPLV

State of practice for debugging leaks ...

Questions containing 'andro | Issues - android - Android | Android Developers Blog: Av | Android Developers Blog: Me

android-developers.blogspot.dk/2011/03/memory-analysis-for-android.html

Android Developers Blog

Developers

SEARCH

Search

ARCHIVE

- 2012 (31)
- 2011 (68)
  - December (7)
  - November (7)
  - October (5)
  - September (5)
  - August (3)
  - July (7)
  - June (3)
  - May (5)

24 MARCH 2011

Memory Analysis for Android Applications

[This post is by Patrick Dubroy, an Android engineer who writes about programming, usability, and interaction on his personal blog. — Tim Bray]

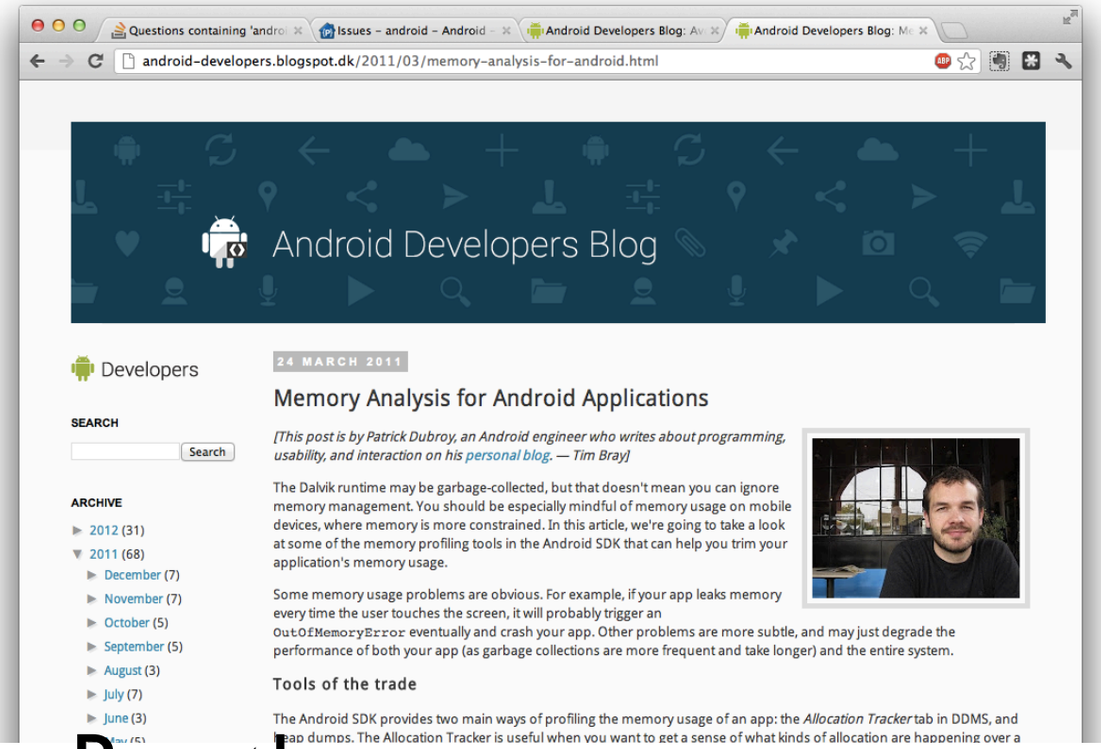
The Dalvik runtime may be garbage-collected, but that doesn't mean you can ignore memory management. You should be especially mindful of memory usage on mobile devices, where memory is more constrained. In this article, we're going to take a look at some of the memory profiling tools in the Android SDK that can help you trim your application's memory usage.

Some memory usage problems are obvious. For example, if your app leaks memory every time the user touches the screen, it will probably trigger an `OutOfMemoryError` eventually and crash your app. Other problems are more subtle, and may just degrade the performance of both your app (as garbage collections are more frequent and take longer) and the entire system.

Tools of the trade

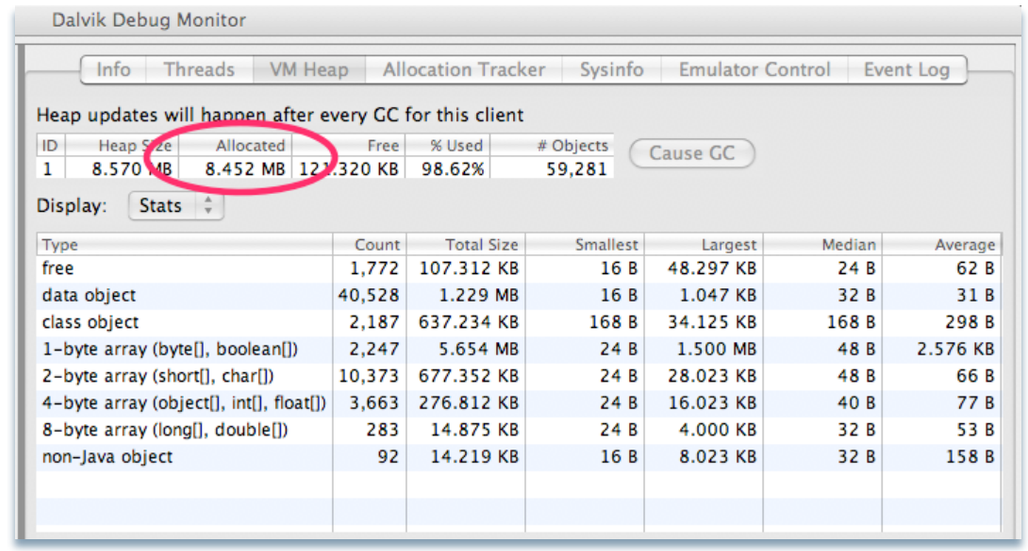
The Android SDK provides two main ways of profiling the memory usage of an app: the *Allocation Tracker* tab in DDMS, and heap dumps. The Allocation Tracker is useful when you want to get a sense of what kinds of allocation are happening over a

State of practice for debugging leaks ...



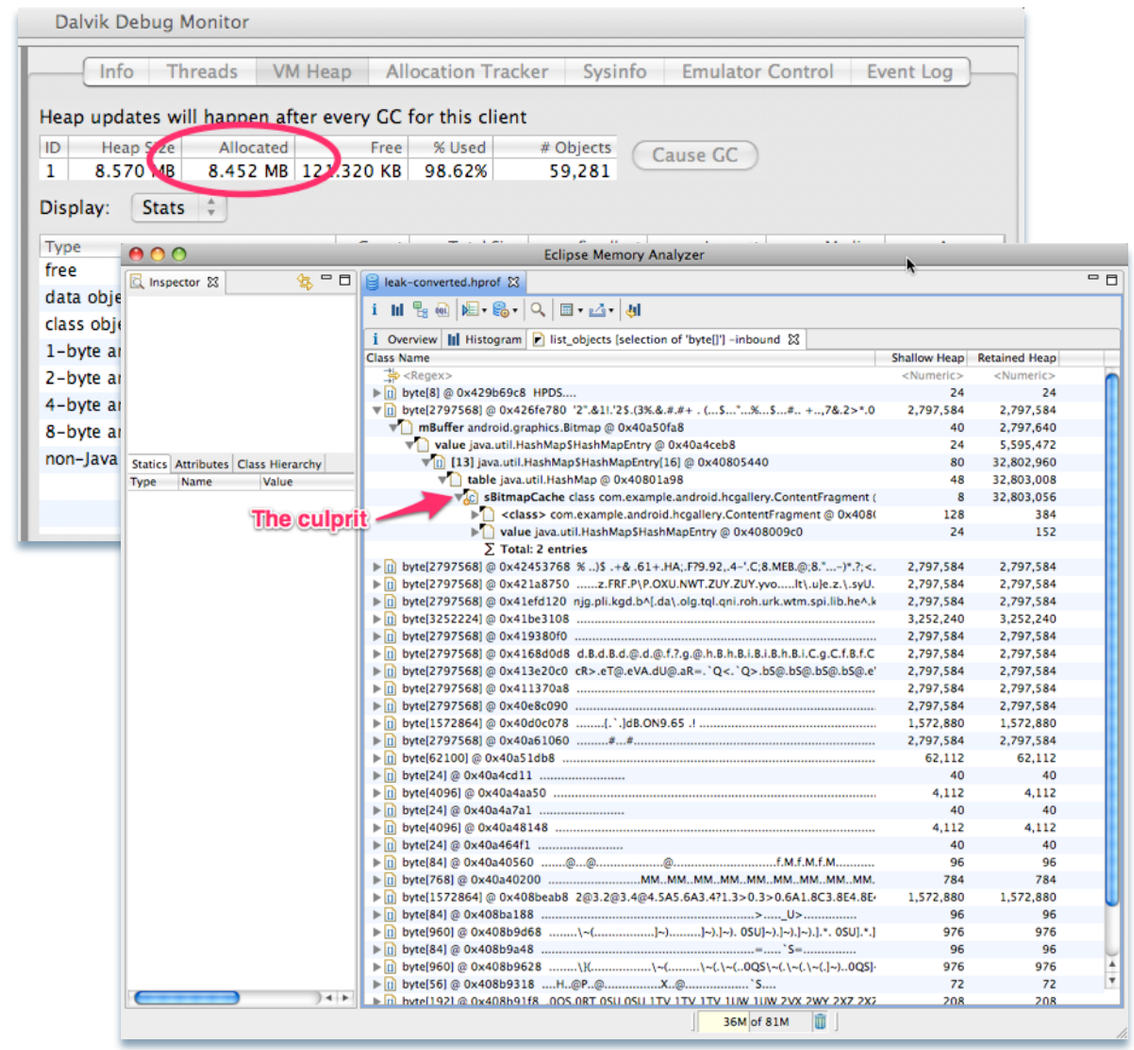
1. Run the app

State of practice for debugging leaks ...



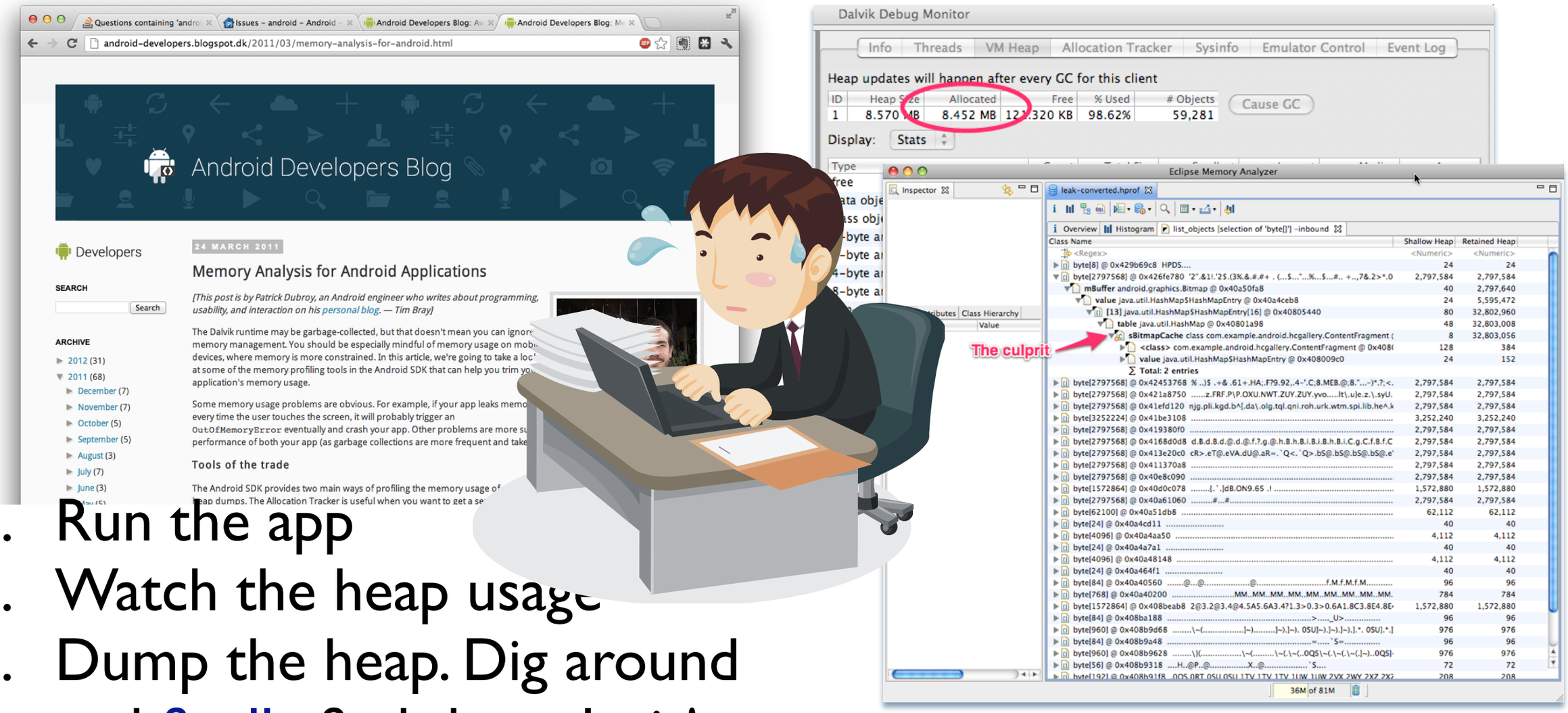
1. Run the app
2. Watch the heap usage

State of practice for debugging leaks ...



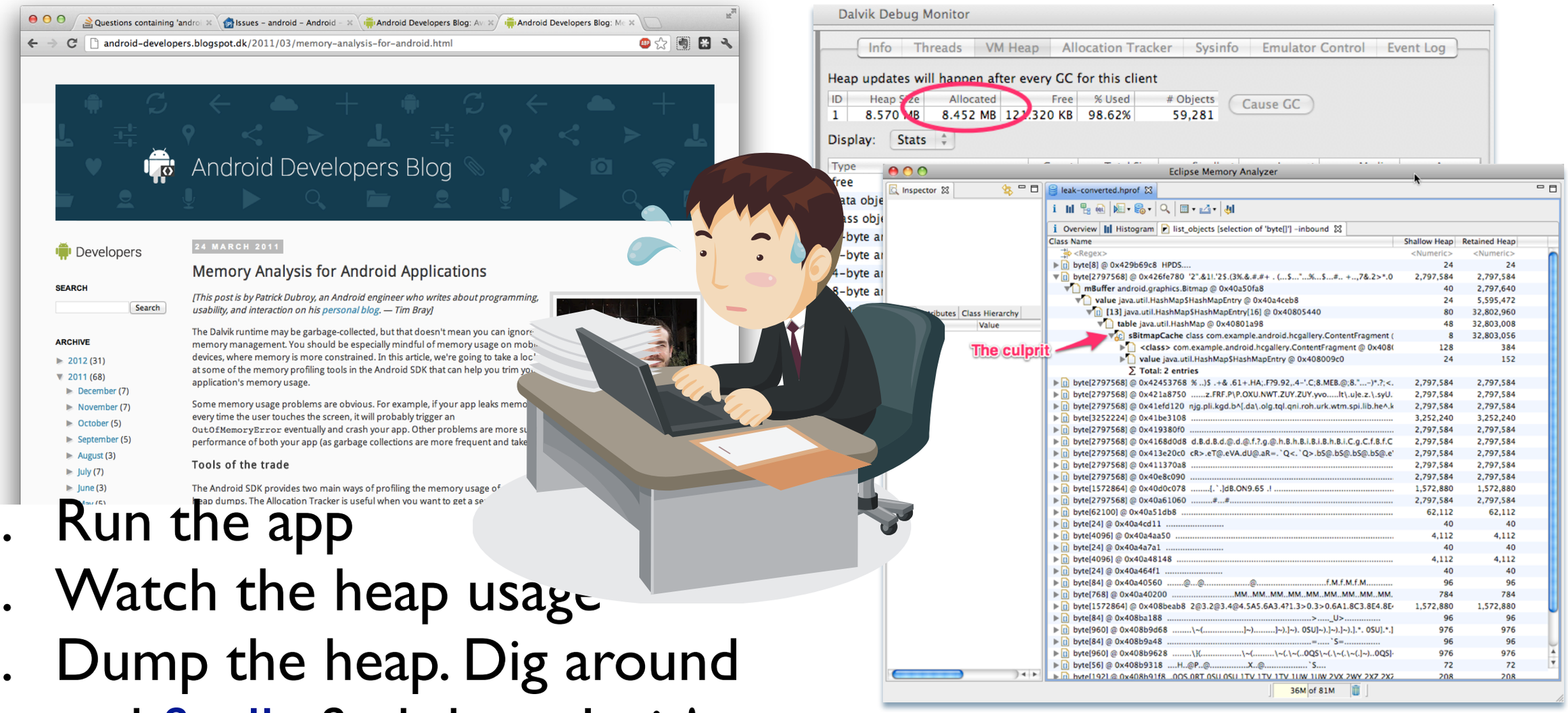
1. Run the app
2. Watch the heap usage
3. Dump the heap. Dig around and finally find the culprit!

State of practice for debugging leaks ...



1. Run the app
2. Watch the heap usage
3. Dump the heap. Dig around and finally find the culprit!

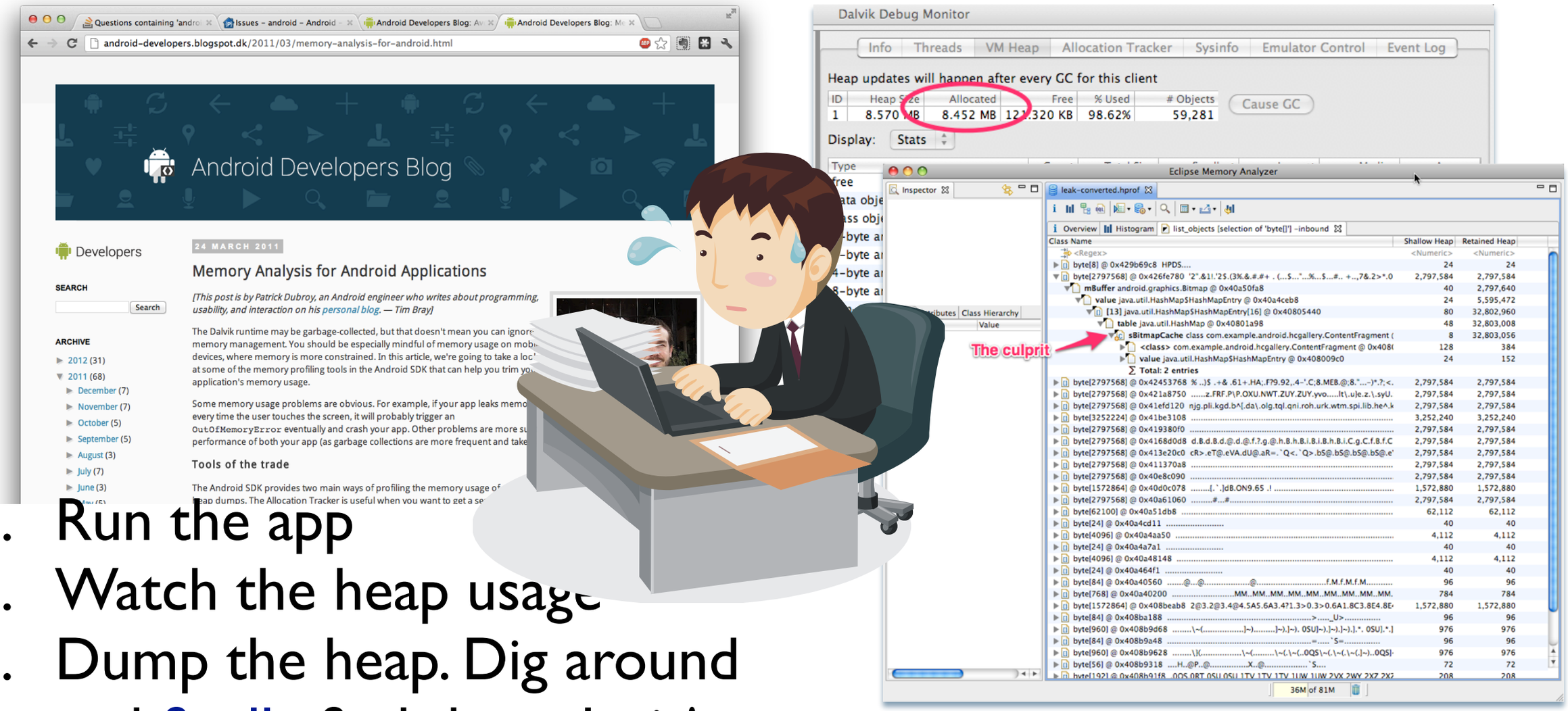
State of practice for debugging leaks ...



1. Run the app
2. Watch the heap usage
3. Dump the heap. Dig around and finally find the culprit!
4. Commit a bugfix

GitHub

State of practice for debugging leaks ...



1. Run the app
2. Watch the heap usage
3. Dump the heap. Dig around and finally find the culprit!
4. Commit a bugfix
5. Bugfix is picked up by our tool **Fixr**

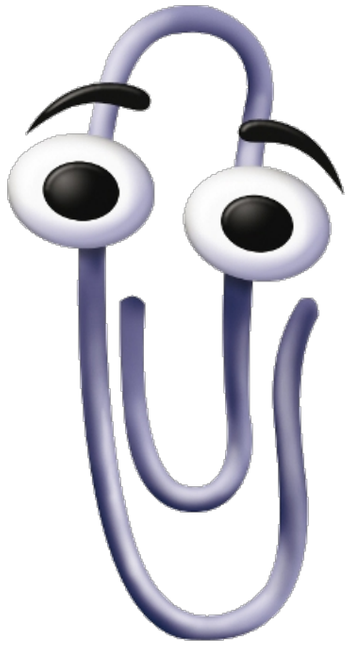
Idea

GitHub

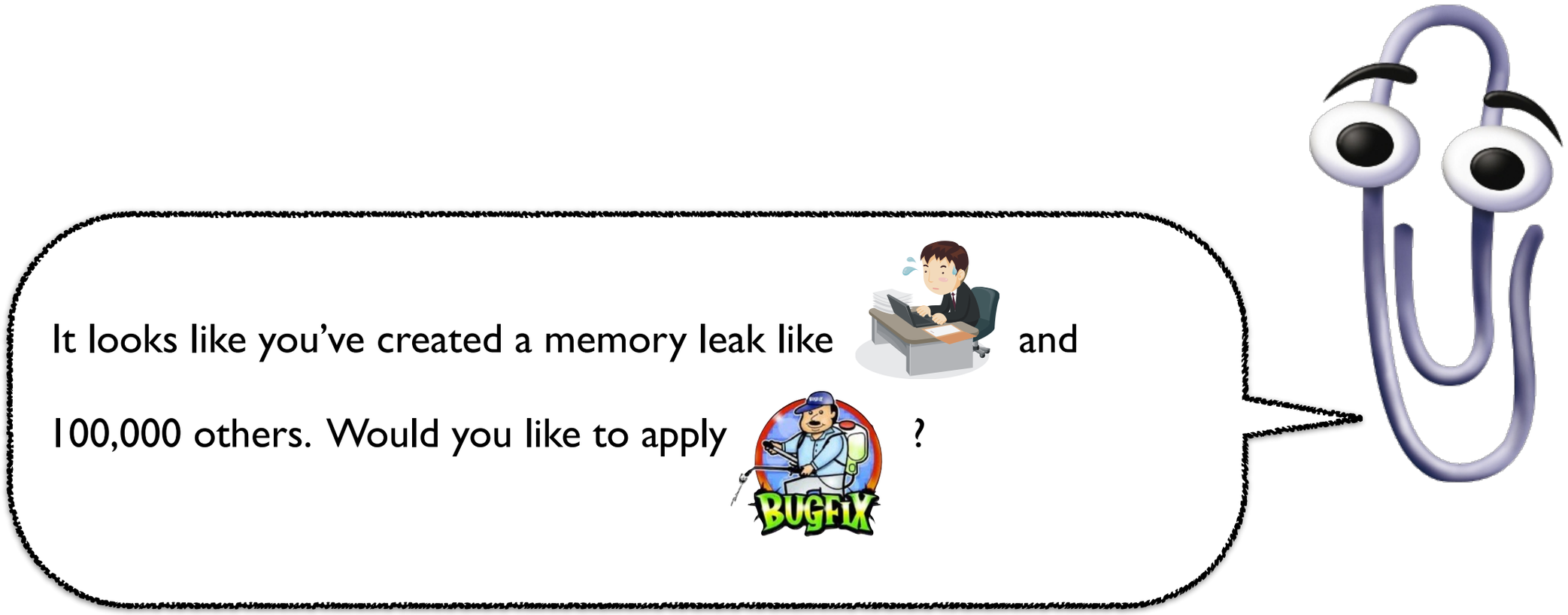
Fixr

A **Fixr**-enabled IDE responds ...

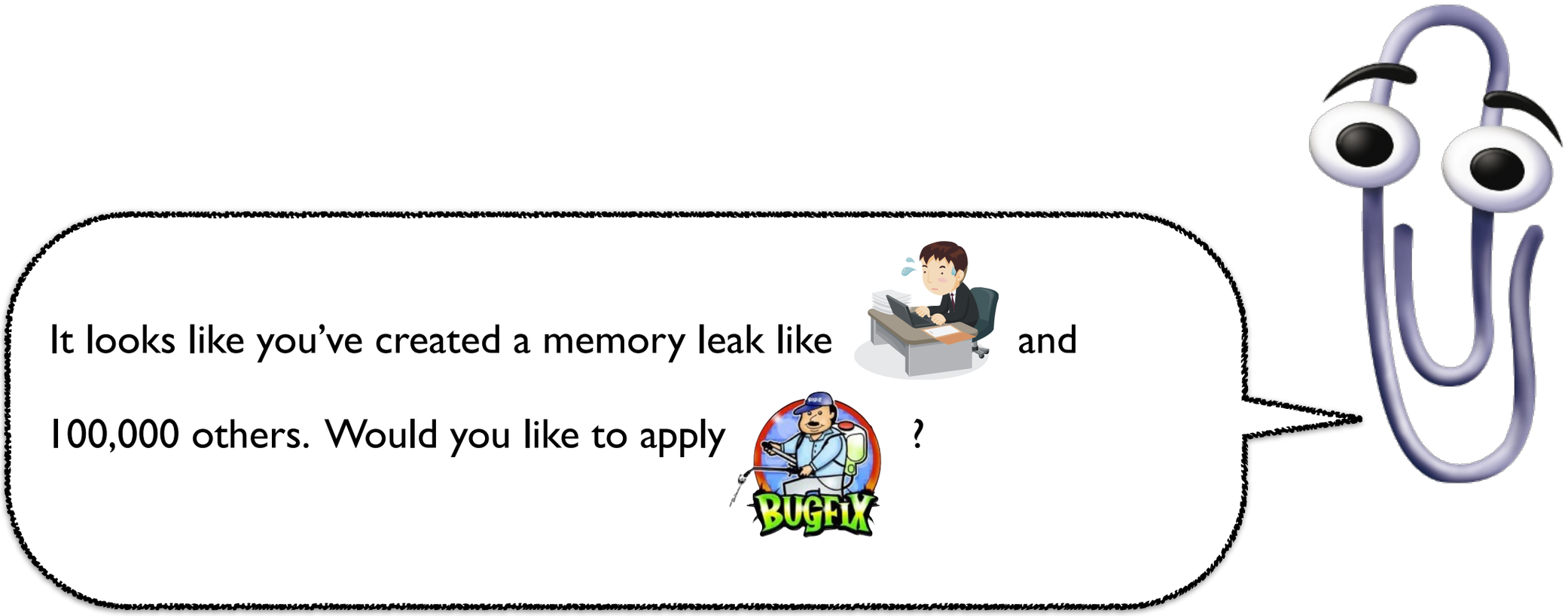
A **Fixr**-enabled IDE responds ...



A **Fixr**-enabled IDE responds …



A **Fixr**-enabled IDE responds …



the bugfix is "transferred"

Program synthesis for network updates

Program synthesis for network updates

Program synthesis

Program synthesis for network updates

Program synthesis

Network Updates

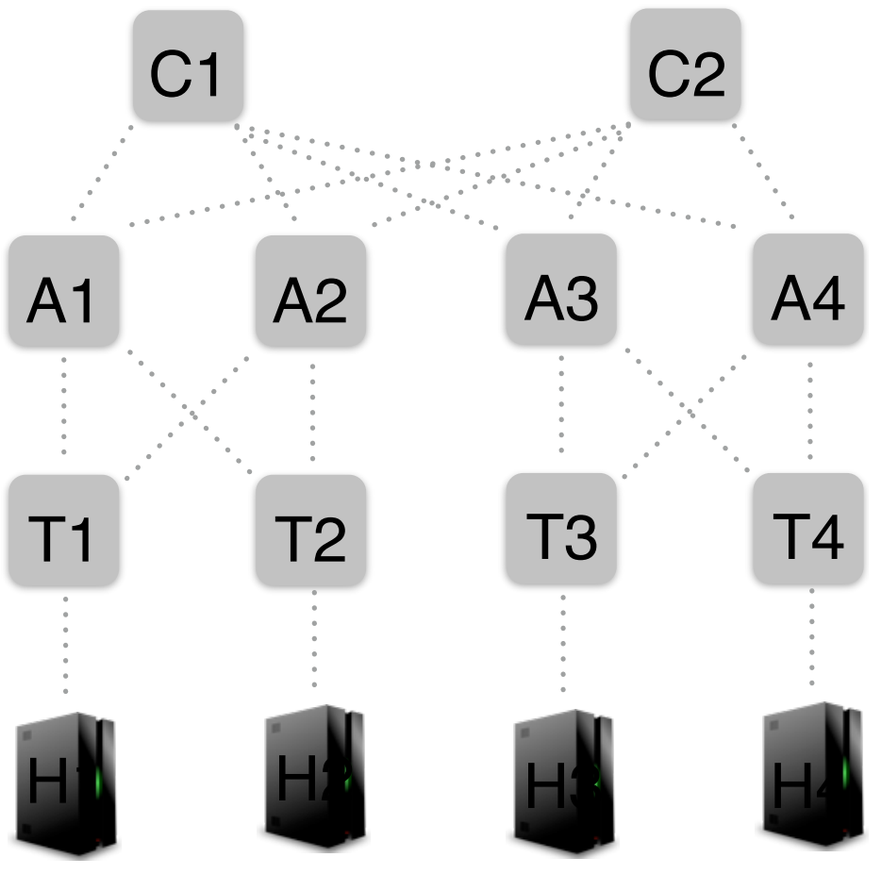
Program synthesis for network updates

Program synthesis

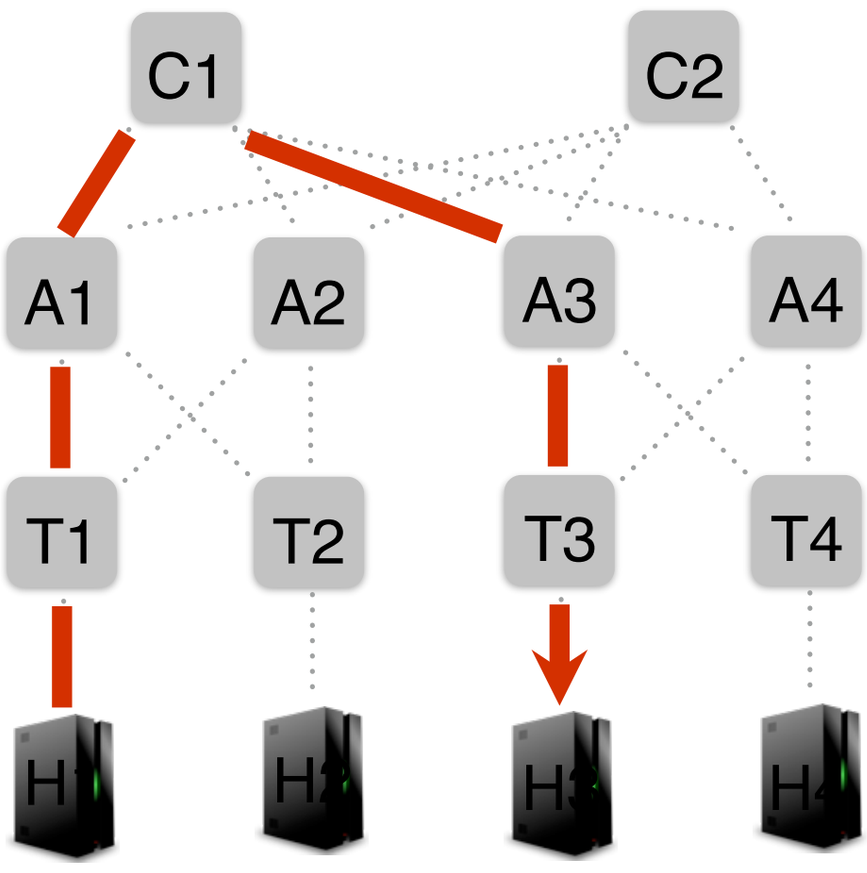
Network Updates

Work to appear at PLDI 2015
First author: Jed McClurg
(second-year PhD student)

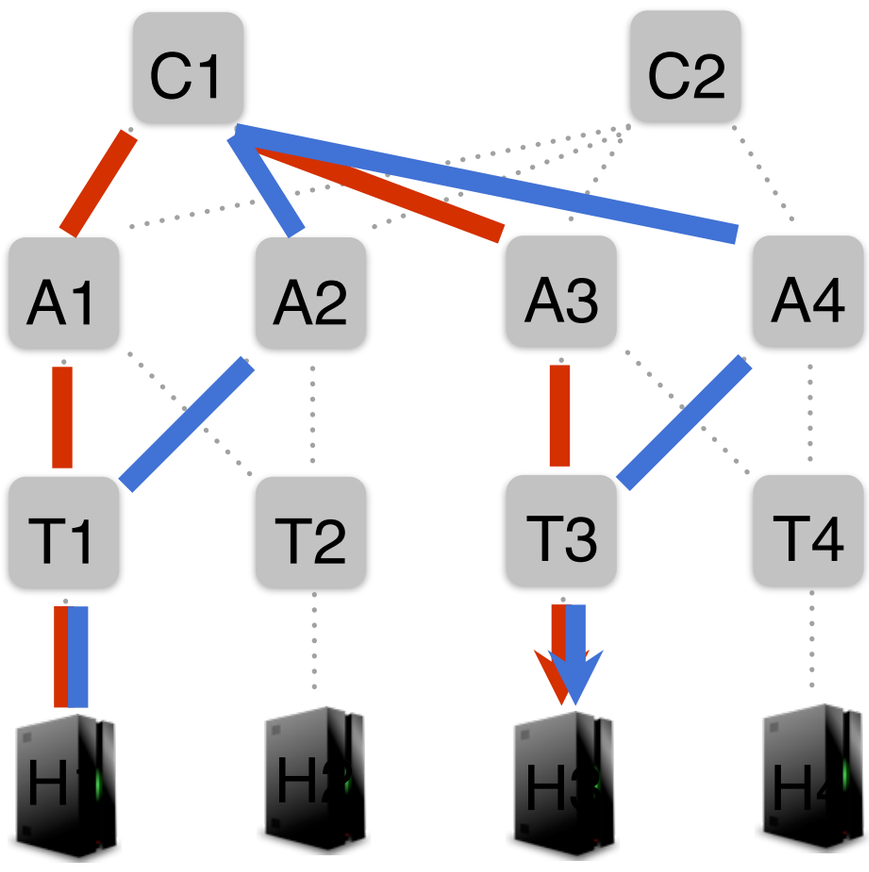
Order Update Example



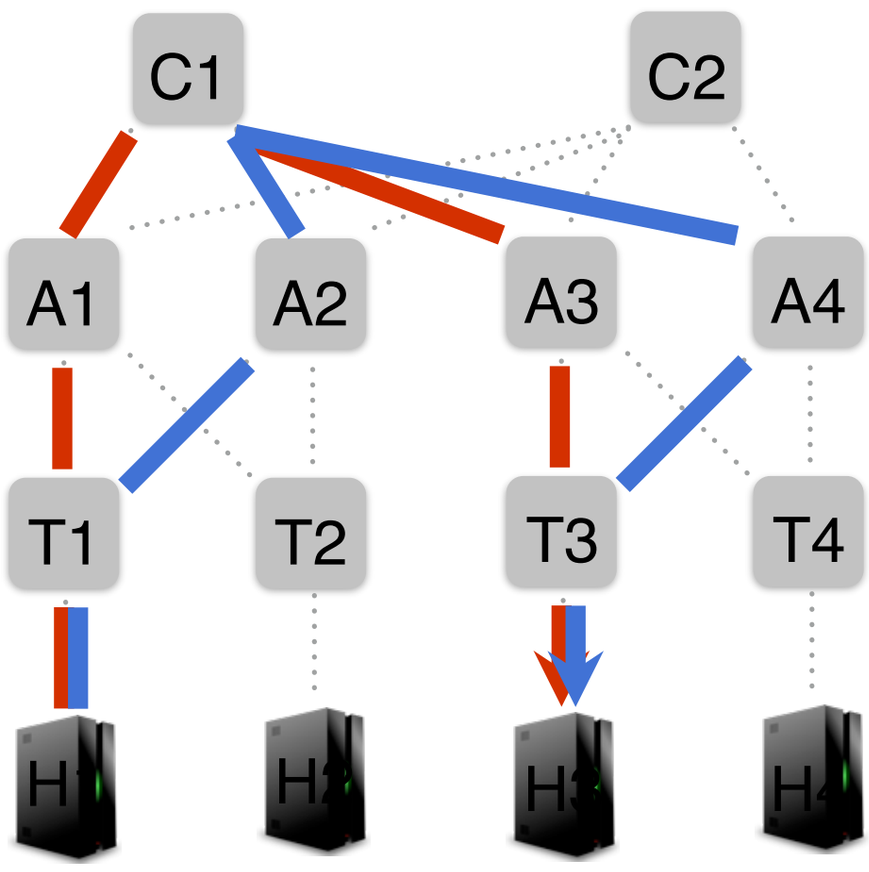
Order Update Example



Order Update Example

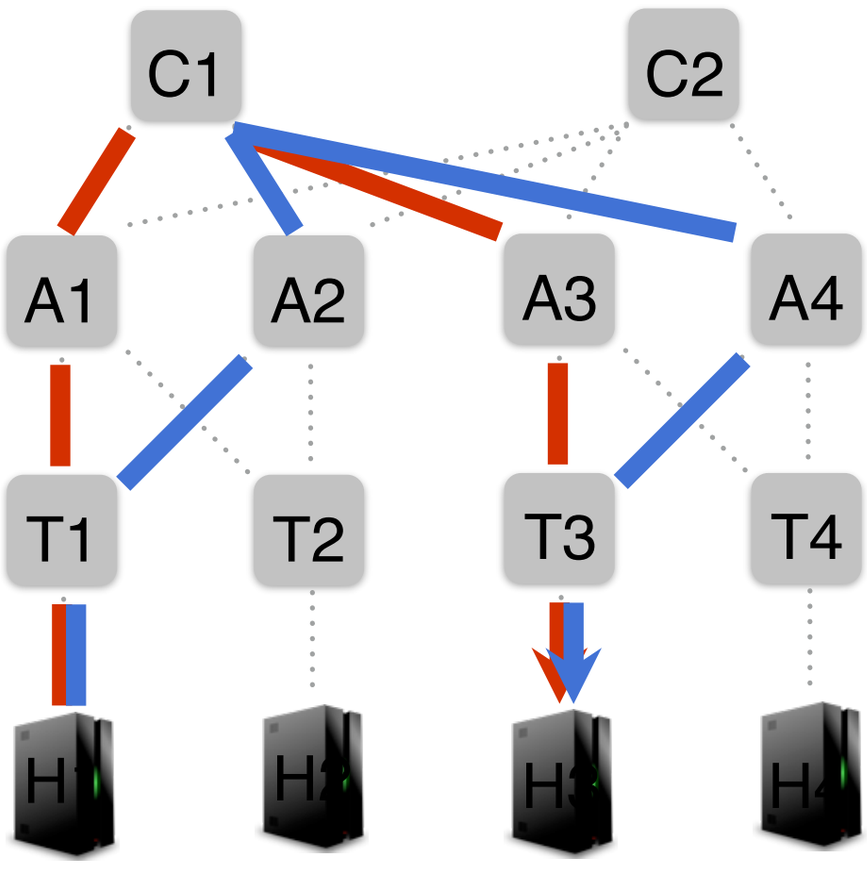


Order Update Example



Property: at all times, maintain H1-H3 connectivity and either traverse A2 or A3

Order Update Example

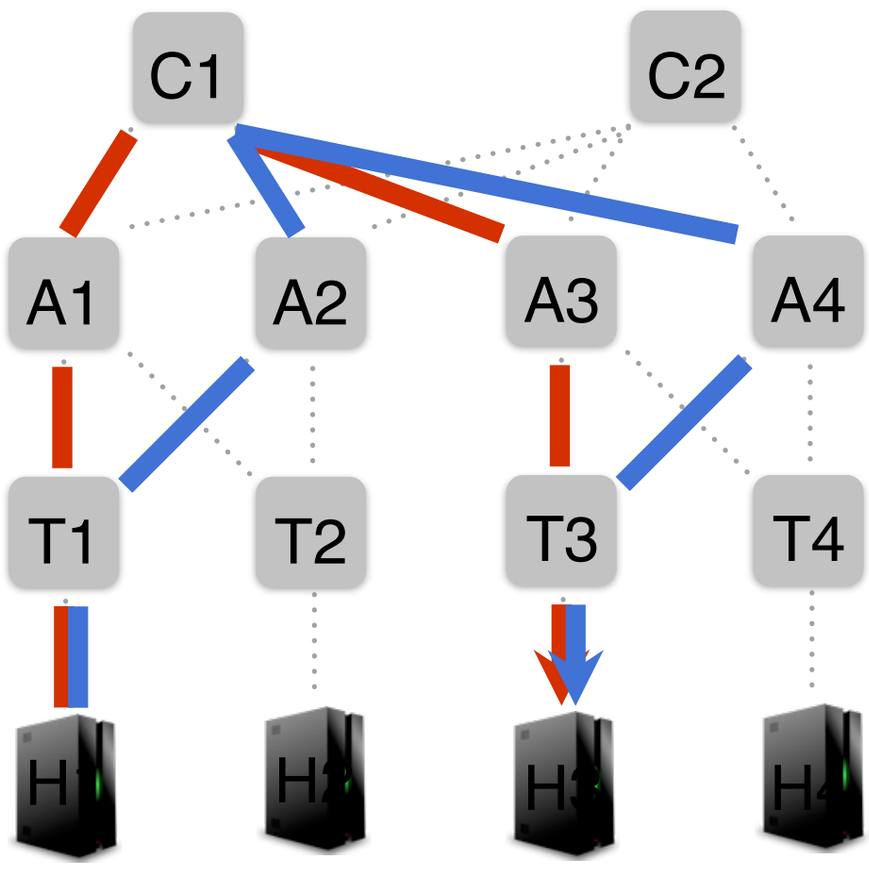


A2-A4-C1 (not good)

A2-A4-T1-C1 ?

Property: at all times, maintain H1-H3 connectivity and either traverse A2 or A3

Order Update Example



A2-A4-C1 (not good)

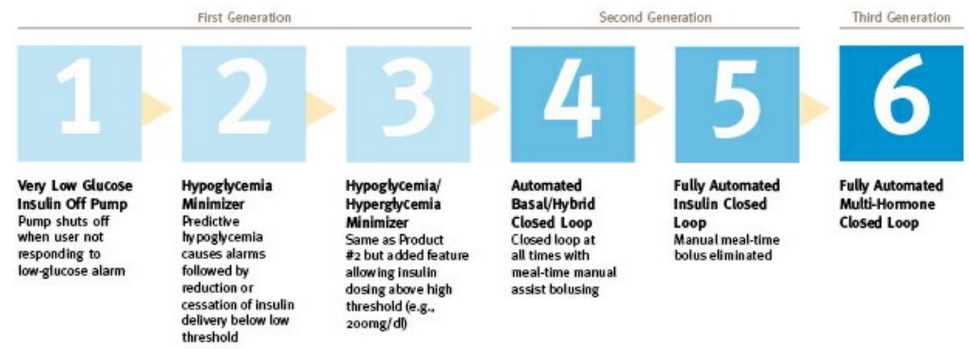
A2-A4-T1-C1 ?

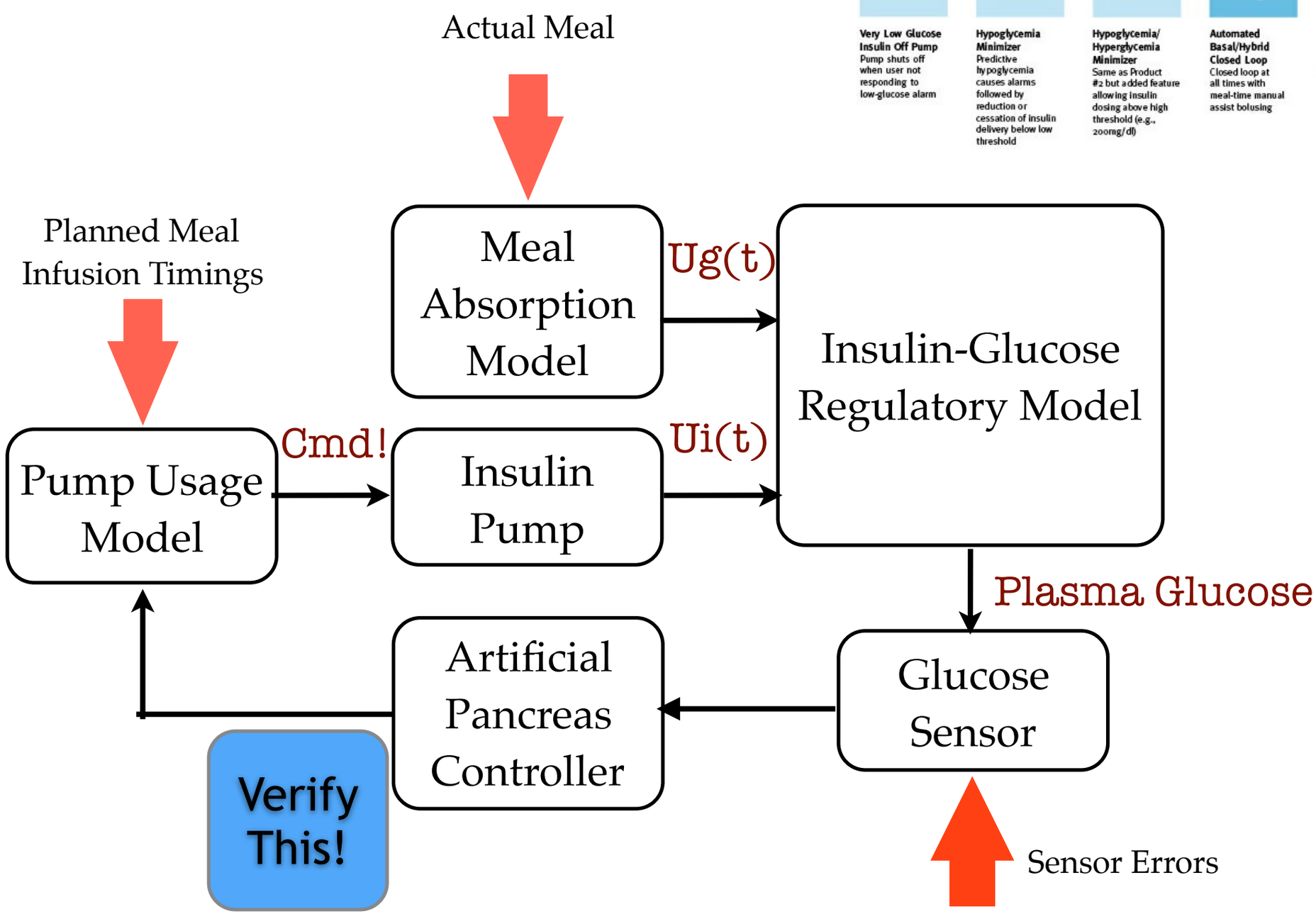
A2-A4-T1-wait-C1

Property: at all times, maintain H1-H3 connectivity and either traverse A2 or A3

Artificial Pancreas Verification Project

Collaboration with UC Denver Medical School and UT El Paso.





PLV research at CU is *successful*!

PLDI 2015: Portland, OR

McClurg, Hojjat, Cerny, Foster. *Efficient Synthesis of Network Updates.*

PLDI 2014: Edinburgh, UK

Logozzo, Fahndrich, Lahiri, Blackshear. *Verification Modulo Versions: Towards Usable Verification.*

POPL 2014 (2): San Diego, CA

Coughlin, Chang. *Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants.*

Jeannet, Schrammel, and Sankaranarayanan. *Abstract Acceleration of General Linear Loops.*

CAV 2014: Vienna, Austria

Cox, Chang, Sankaranarayanan. *QUICr: A Reusable Library for Parametric Abstraction of Sets and Numbers.*

ESOP 2015 (2): London, UK

Cox, Chang, Rival. *Desynchronized Multi-State Abstractions for Open Programs in Dynamic Languages.*

Cerny, Henzinger, Kovacs, Radhakrishna, Zwirchmayr. *Segment Abstraction for Worst-Case Execution Time Analysis.*

EMSOFT 2014 (2): New Delhi, India

Ravanbakhsh, Sankaranarayanan. *Infinite Horizon Safety Controller Synthesis through Disjunctive Polyhedral Abstract Interpretation.*

Zutshi, Sankaranarayanan, Deshmukh, Kapinski. *Multiple Shooting, CEGAR-based Falsification for Hybrid Systems.*

PLV research at CU is *successful*!

PLDI 2015: Portland, OR

McClurg, Hojjat, Cerny, Foster. *Efficient Synthesis of Network Updates.*

PLDI 2014: Edinburgh, UK

Logozzo, Fahndrich, Lahiri, Blackshear. *Verification Modulo Versions: Towards Usable Verification.*

POPL 2014 (2): San Diego, CA

Coughlin, Chang. *Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants.*
Jeannet, Schrammel, and Sankaranarayanan. *Abstract Acceleration of General Linear Loops.*

CAV 2014: Vienna, Austria

Cox, Chang, Sankaranarayanan. *QUIC* ... *ric Abstraction of Sets and Numbers.*

Fun Destinations!

ESOP 2015 (2): Lo

Cox, Chang, Rival. *Desynchronized Multi-State Abstractions for Open Programs in Dynamic Languages.*
Cerny, Henzinger, Kovacs, Radhakrishna, Zwirchmayr. *Segment Abstraction for Worst-Case Execution Time Analysis.*

EMSOFT 2014 (2): New Delhi, India

Ravanbakhsh, Sankaranarayanan. *Infinite Horizon Safety Controller Synthesis through Disjunctive Polyhedral Abstract Interpretation.*

Zutshi, Sankaranarayanan, Deshmukh, Kapinski. *Multiple Shooting, CEGAR-based Falsification for Hybrid Systems.*

Top Venues!

arch at CU is *successful*!

PLDI 2015: Portland, OR

McClurg, Hojjat, Cerny, Foster. *Efficient Synthesis of Network Updates.*

PLDI 2014: Edinburgh, UK

Logozzo, Fahndrich, Lahiri, Blackshear. *Verification Modulo Versions: Towards Usable Verification.*

POPL 2014 (2): San Diego, CA

Coughlin, Chang. *Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants.*

Jeannet, Schrammel, and Sankaranarayanan. *Abstract Acceleration of General Linear Loops.*

CAV 2014: Vienna, Austria

Cox, Chang, Sankaranarayanan. *QUIC ric Abstraction of Sets and Numbers.*

Fun Destinations!

ESOP 2015 (2): Lo

Cox, Chang, Rival. *Desynchronized Multi-State Abstractions for Open Programs in Dynamic Languages.*

Cerny, Henzinger, Kovacs, Radhakrishna, Zwirchmayr. *Segment Abstraction for Worst-Case Execution Time Analysis.*

EMSOFT 2014 (2): New Delhi, India

Ravanbakhsh, Sankaranarayanan. *Infinite Horizon Safety Controller Synthesis through Disjunctive Polyhedral Abstract Interpretation.*

Zutshi, Sankaranarayanan, Deshmukh, Kapinski. *Multiple Shooting, CEGAR-based Falsification for Hybrid Systems.*

Top Venues!

arch at CU is *successful*!

PLDI 2015: Portland, OR

McClurg, Hojjat, Cerny, Foster. *Efficient Synthesis of Network Updates.*

PLDI 2014: Edinburgh, UK

Logozzo, Fahndrich, Lahiri, Blackshear. *Verification Modulo Versions: Towards Usable Verification.*

POPL 2014 (2): San Diego, CA

Coughlin, Chang. *Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants.*

Jeannet, Schrammel, and Sankaranarayanan. *Abstract Acceleration of General Linear Loops.*

CAV 2014: Vienna, Austria

Cox, Chang, Sankaranarayanan. *QUIC[illegible]ric Abstraction of Sets and Numbers.*

Fun Destinations!

ESOP 2015 (2): Lo[illegible]

Cox, Chang, Rival. *Desynchronized Multi-State Abstractions for Open Programs in Dynamic Languages.*

Cerny, Henzinger, Kovacs, Radhakrishna, Zwirchmayr. *Segment Abstraction for Worst-Case Execution Time Analysis.*

EMSOFT 2014 (2): New Delhi, India

Ravanbakhsh, Sankaranarayanan. *Infinite Horizon Safety Controller Synthesis through Disjunctive Polyhedral Abstract Interpretation.*

Zutshi, Sankaranarayanan, Deshmukh, Kapinski. *Multiple Shooting, CEGAR-based Falsification for Hybrid Systems.*

Top Venues!

arch at CU is *successful*!

PLDI 2015: Portland, OR

McClurg, Hojjat, Cerny, Foster. *Efficient Synthesis of Network Updates.*

PLDI 2014: Edinburgh, UK

Logozzo, Fahndrich, Lahiri, Blackshear. *Verification Modulo Versions: Towards Usable Verification.*

POPL 2014 (2): San Diego, CA

Coughlin, Chang. *Fissile Type Analysis: Modular Checking of Almost Everywhere Invariants.*

Jeannet, Schrammel, and Sankaranarayanan. *Abstract Acceleration of General Linear Loops.*

CAV 2014: Vienna, Austria

Cox, Chang, Sankaranarayanan. *QUIC[illegible]ric Abstraction of Sets and Numbers.*

Fun Destinations!

ESOP 2015 (2): Lo[illegible]

Cox, Chang, Rival. *Desynchronized Multi-State Abstractions for Open Programs in Dynamic Languages.*

Cerny, Henzinger, Kovacs, Radhakrishna, Zwirchmayr. *Segment Abstraction for Worst-Case Execution Time Analysis.*

EMSOFT 2014 (2): New Delhi, India

Ravanbakhsh, Sankaranarayanan. *Infinite Horizon Safety Controller Synthesis through Disjunctive Polyhedral Abstract Interpretation.*

Zutshi, Sankaranarayanan, Deshmukh, Kapinski. *Multiple Shooting, CEGAR-based Falsification for Hybrid Systems.*

CUPLV Students!

PLV research at CU has *world-wide collaborations*!

PLV research at CU has *world-wide collaborations*!

Collaborators ⇒
Internships and Jobs

PLV students have *interned* at …

Microsoft Research

Redmond, Washington
Cambridge, UK
Bangalore, India

Google

IBM

facebook

NEC

parc
Palo Alto Research Center

SRI International

TOYOTA
TOYOTA TECHNICAL CENTER

CRAY
THE SUPERCOMPUTER COMPANY

The PLV group has *fun* together!

The PLV group has *fun* together!

Group meetings at the Boulder Tea House twice a semester

The PLV group has *fun* together!

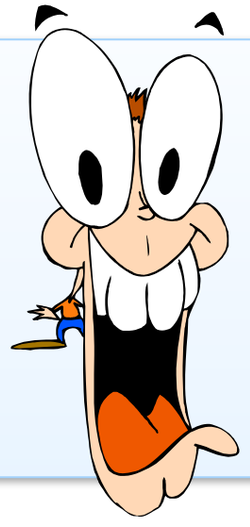
Group meetings at the Boulder Tea House twice a semester

Travel to conferences (POPL 2012)

The PLV group has *fun* together!

Group meetings at the Boulder Tea House twice a semester

Travel to conferences (POPL 2012)

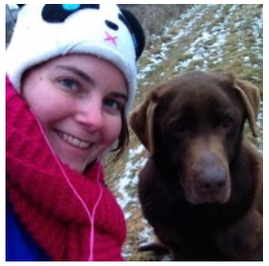


Our mentoring: Guide you to research that excites you!

Our group


Shawn


Jed


Alex

Post-Doc


Vris


Amin

BS


Evan


Kyle


Max

PhD


Sam


Aleks


Hadi


Aditya


Fabio


Sriram


You?


You?

Faculty


Pavol


Evan


Dirk

Some of our other research projects

- False alarm triage analysis
- Modular invariant checking
- Analysis of dynamic languages
- Mobile app malware detection
- Incremental verification-validation
- Analysis of medical devices
- Health care process analysis
- Cyber-physical systems verification
- Program synthesis
- Synthesis for software-defined networking

- **And soon projects created by you!**