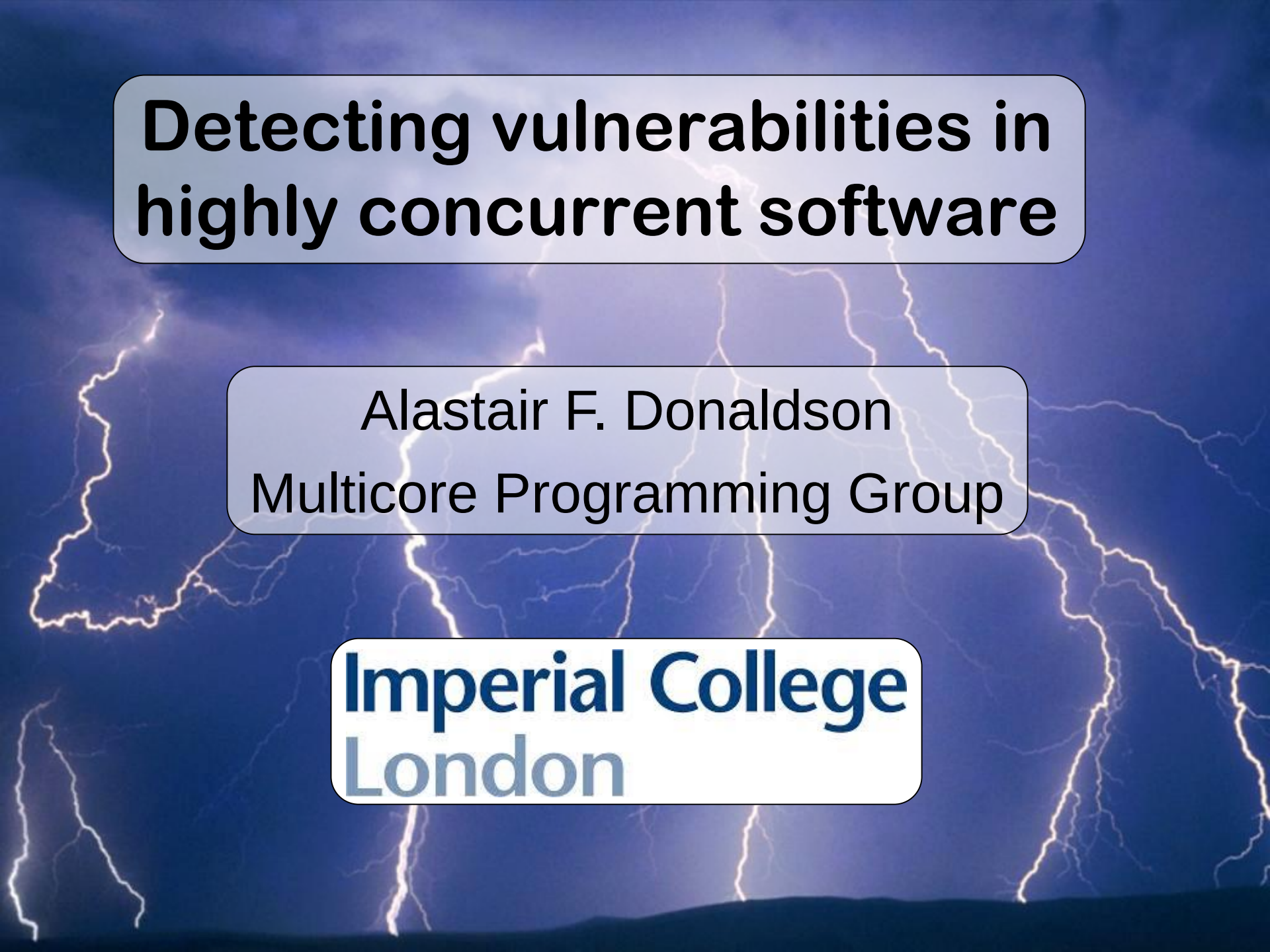


Detecting vulnerabilities in highly concurrent software

Alastair F. Donaldson
Multicore Programming Group

Imperial College London

Analysis, Verification and Testing at Imperial (INVEST)

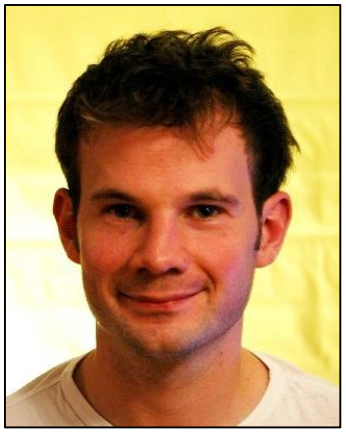
Philippa

Reasoning about web and concurrent programs

Cristian

Practical techniques for improving SW reliability and security

Ally



Automated program analysis for multi-/ many-core software

Part of **Programming Languages and Systems** at Imperial

More broadly: part of buzzing "London-Cambridge-Oxford" Programming Languages community

Multicore Programming Group

GPU software verification


Nathan


Jeroen


Dan

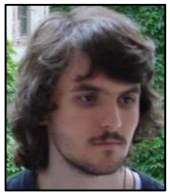

Adam

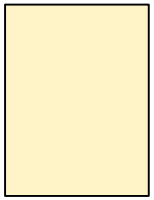
Systematic Concurrency Testing


Paul


Ethel

Many-core compiler validation


Andrei

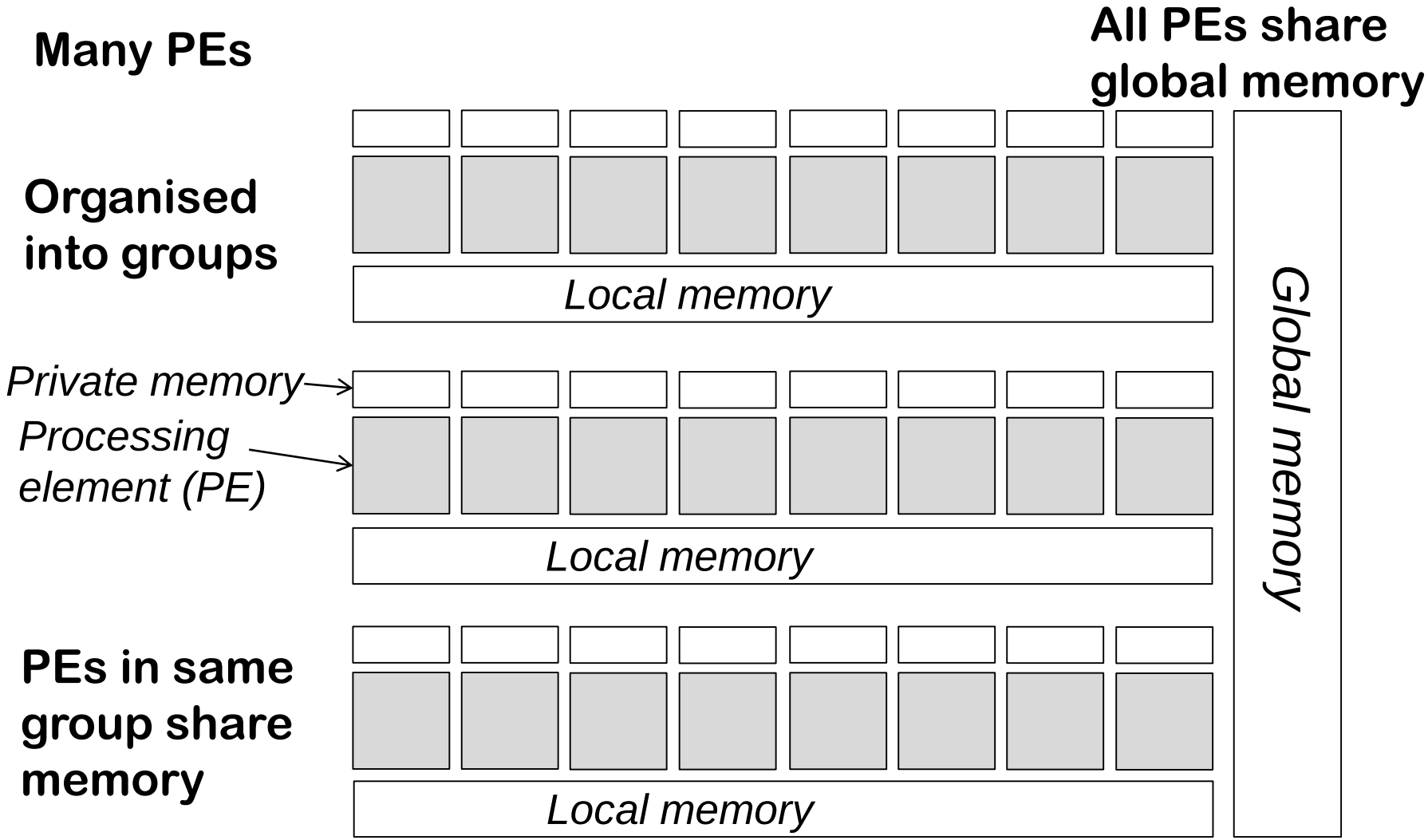

Chris

Programming language and memory models

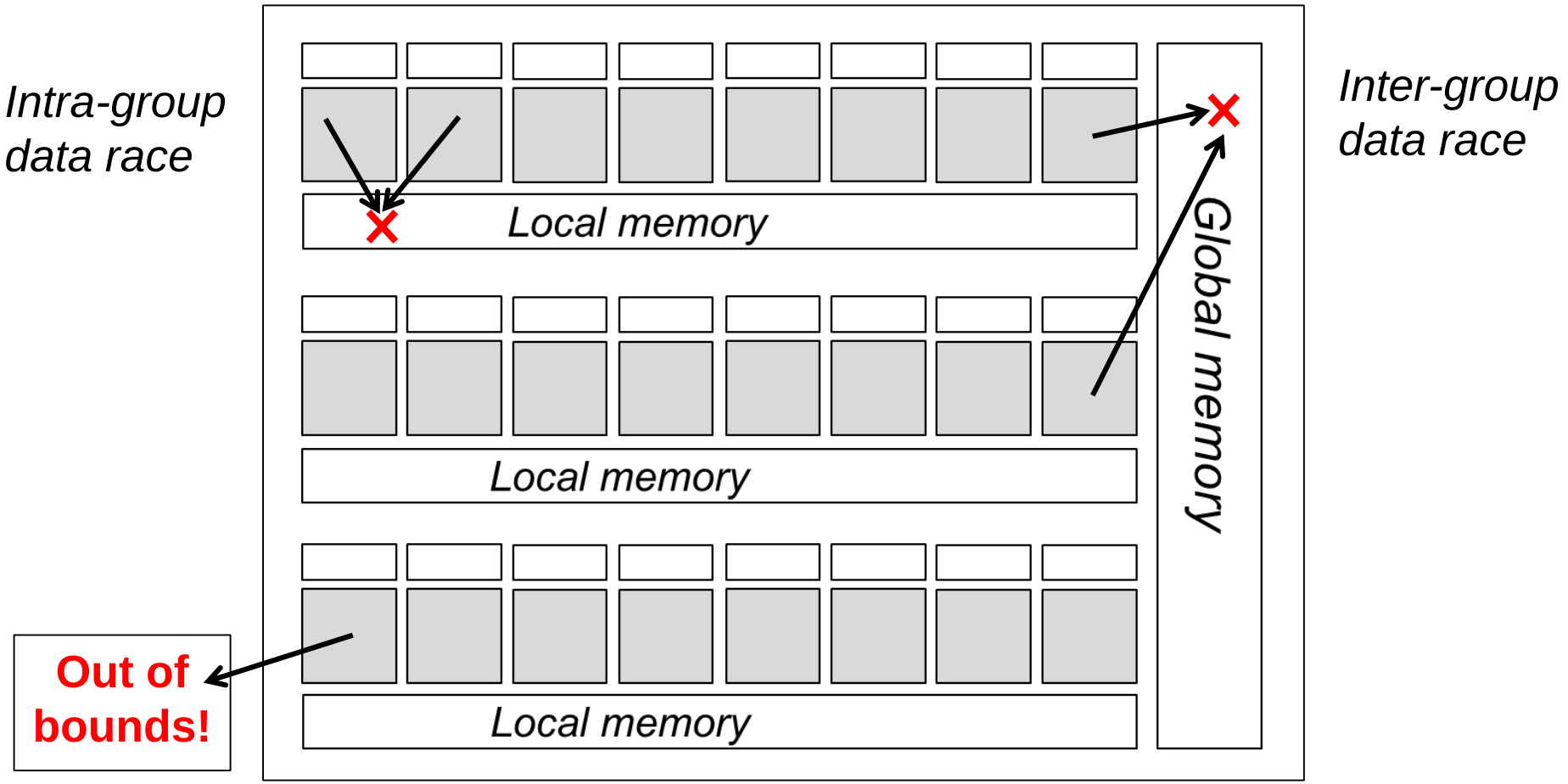

John


Pantazis

Many-core processors: GPUs



Memory safety errors in parallel programs



Errors can be security critical as GPUs are widely deployed

WebCL and **RenderScript** – emerging standards for GPUs in **web** and **embedded** programs

GPUVerify: static verification for GPU kernels

Leverages dramatic advances in **static verification** to:

- Find **defects**
- Prove **absence of defects**

in **many-core** software

Industrial support from: ARM, Microsoft Research

Industrial impact:

- ARM Mali Graphics debugger
- Imagination 3rd party showcase
- AMD, NVIDIA Identified software defects

Activities to aid transfer to industry

Engineering (it really works)

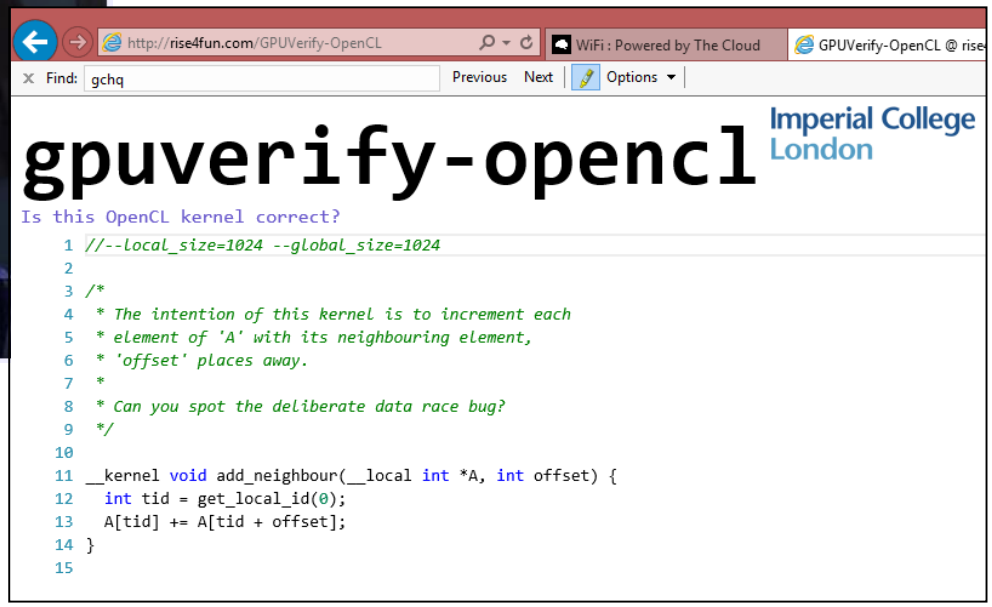
Supported by

EPSRC

Pathways to Impact project

Video tutorials

Web interface for tools



Community recognition for tech transfer

HiPEAC COMPILATION ARCHITECTURE

The Steering Committee of the
HiPEAC Network of Excellence
has awarded a

SEVENTH FRAMEWORK PROGRAMME

HiPEAC Technology Transfer Award

to

The Multicore Programming Group, Imperial College London

for a technology transfer on

GPUVerify

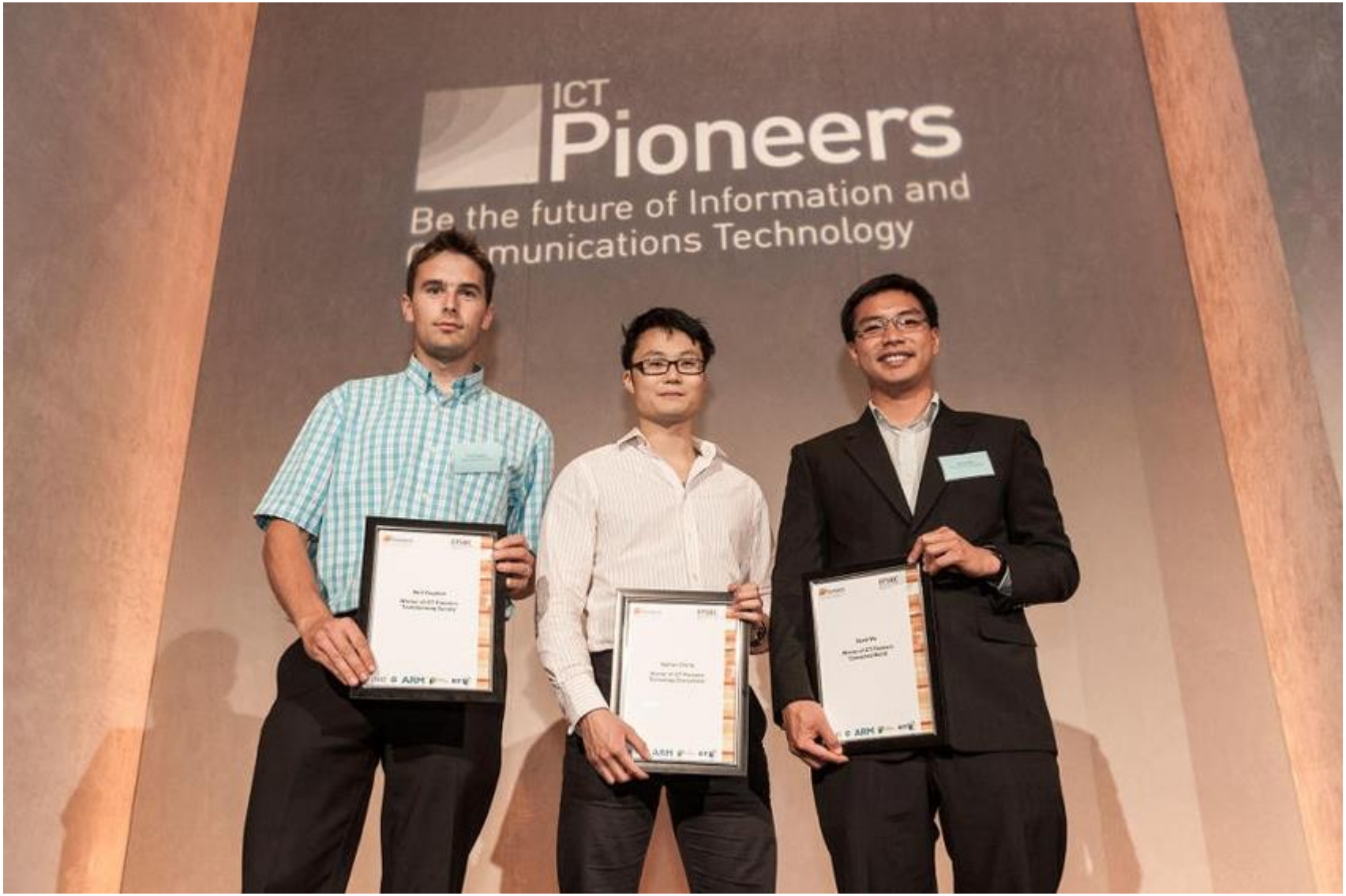
with

ARM

The HiPEAC Coordinator

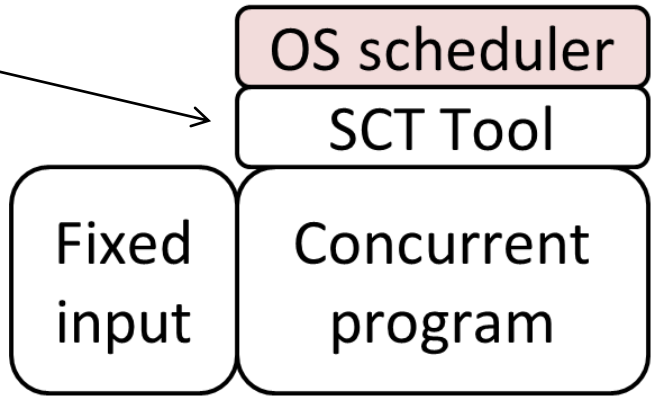
Prof. Koen De Bosschere

Community recognition for tech transfer

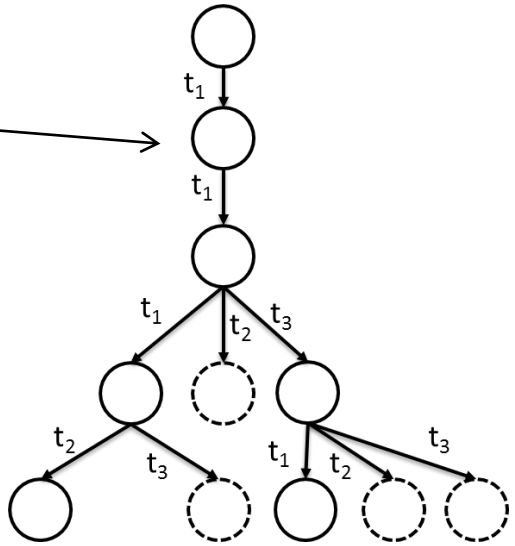


Systematic concurrency testing

Intercept OS scheduler

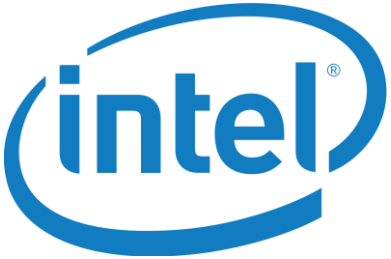


Explore schedule space



Apply **state-space reduction** and **schedule prioritization** to find bugs

Systematic concurrency testing: engagement with industry / government



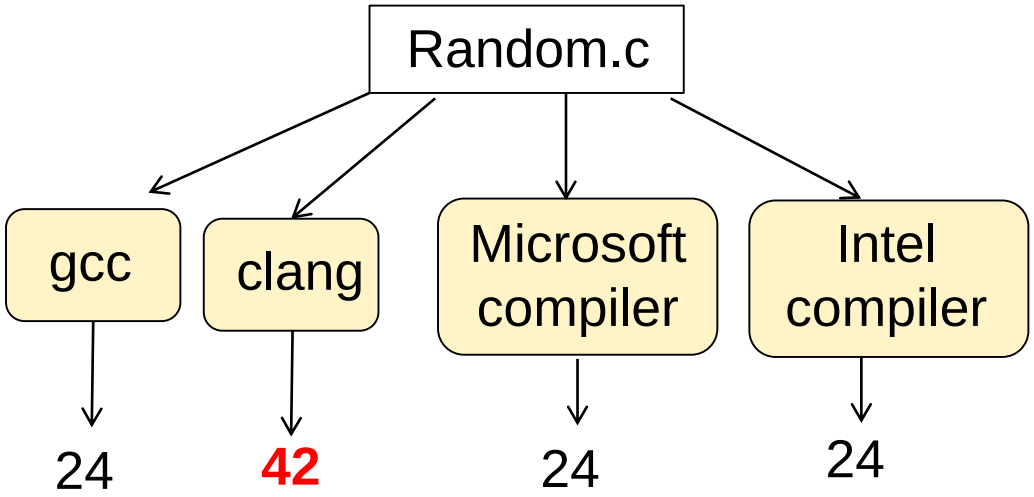
High assurance device drivers
(2 funded PhD students)

Security vulnerabilities in concurrent software (PhD funding)

Gave recent talk about our work, following up on **dynamic analysis of concurrent Android apps**

Compiler vulnerability analysis

Random differential testing



Pioneered by **Csmith**, University of Utah

We are lifting this idea to
many-core compilers

Discovered defects so far in GPU compilers from all major vendors

Many-core memory models

Not enough time to talk about it!

We have discovered issues with:

OpenCL 2.0 memory model

Published code examples

Compilers from main vendors

GPU chips

With:

- Jade Alglave (UCL)
- Mark Batty (Cambridge)
- Ganesh Gopalakrishnan (Utah)
- Jeroen Ketema (Imperial)
- Daniel Poetzl (Oxford)
- Tyler Sorensen (UCL)
- John Wickerson (Imperial)

All could lead to security vulnerabilities

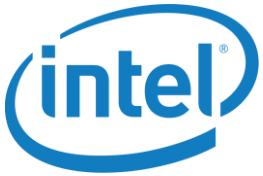
Interacting with KHRONOS GROUP HSA FOUNDATION

Builds on pioneering C/C++11 work by various authors including Mark Batty and **Peter Sewell** (Cambridge)

Want to know more? Get in touch!

alastair.donaldson@imperial.ac.uk

Funding support



Industrial Collaborators

AMD, ARM, Codeplay, Facebook, Imagination, Intel, Microsoft Research, NVIDIA, Realeyes, Rightware

Multicore Programming Group @ Imperial

Ethel Bardsley, Adam Betts, Nathan Chong, Pantazis Deligiannis, Jeroen Ketema, Andrei Lascu, Chris Lidbury, Dan Liew, Paul Thomson, John Wickerson,

Academic Collaborators

Jade Alglave (UCL), Mark Batty (Cambridge), Albert Cohen (INRIA), Ganesh Gopalakrishnan (Utah), Daniel Kroening (Oxford), Daniel Poetzl (Oxford), Tyler Sorensen (UCL)