Helping Programmers Write Better Programs

Nathan Chong, Imperial College London
nyc04@imperial.ac.uk

1. Technology (Bugs) Everywhere

(a) Therac-25  
(b) Ariane-5
(c) Intel FDIV  
(d) Microsoft Zune
(e) Heartbleed
(f) Petrov Incident
(g) Airbus A380
(h) Toyota ECM
(i) Knight Capital

2. GPUs

Graphics Processing Units are everywhere and used for diverse applications.

3. GPU Bugs

thread 1
\[ x := 1 \]
thread 2
\[ x := 2 \]

\[ ? \]
race!
x is 1 or 2?

Orchestrating many interacting pieces at the same time is challenging. Races introduce undefined behaviour.

4. Formal Verification for GPU Programs

GPU program \[ \rightarrow \]
GPUVerify

Possible race!

No bugs found

Guarantees beyond the scope of manual testing

My research inside:
Four top-tier publications

Key idea: transform program into logical form

5. Technology Transfer: A Real Tool Finding Real Bugs in Real Code

[About a bug found by GPUVerify] It was a real bug, and it caused real issues in the results. It took significant debugging time to find the problem. [GPUVerify found this bug automatically]

Lars Nyland
Senior Architect

We think that OpenCL developers will find GPUVerify useful and we are keen to promote and distribute the tool through our PowerVR developer program.

Carlos Sarria
Senior Developer Technology Manager,
PowerVR Graphics

Following up on Nathan’s work, we are sponsoring another [CASE] PhD student at Imperial College. We are keen for ARM Mali GPU developers to use the tool in their workflow, so are exploring opportunities for integrating it with Mali OpenCL tools.

Anton Lokhmotov
Staff Engineer,
GPU Compute Compiler Team Lead

This work was supported by the EPSRC PSL project (EP/I006761/1), EPSRC first grant (EP/K011499/1) and the EU FP7 STREP project CARP (287767)