TETRACOM: Technology Transfer in Computing Systems

FP7 Coordination and Support Action to fund 50 technology transfer projects (TTP) in computing systems. This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement n° 609491.

CK/CLsmith: An Automated Testing Framework for Many-Core Vendor Tools

Alastair F. Donaldson, Andrei Lascu, Imperial College London, UK
Grigori Fursin, Anton Lokhmotov, dividiti Ltd., UK

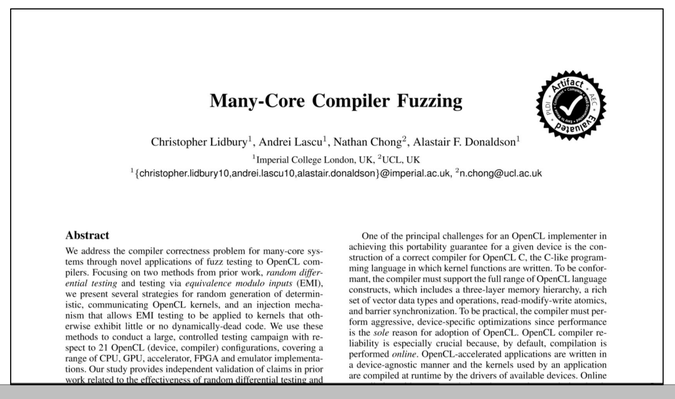
TTP Problem

Many-core APIs: large, sophisticated, hard to implement

Spans CPUs, GPUs, FPGAS

Unites graphics and compute

OpenCL compilers: prone to *wrong code* bugs



Many-Core Compiler Fuzzing

Christopher Lidbury[1], Andrei Lascu[1], Nathan Chong[2], Alastair F. Donaldson[1]
[1]Imperial College London, UK, [2]UCL, UK
[1]{christopher.lidbury10,andrei.lascu10,alastair.donaldson}@imperial.ac.uk, [2]n.chong@ucl.ac.uk

Abstract

We address the compiler correctness problem for many-core systems through novel applications of fuzz testing to OpenCL compilers. Focusing on two methods from prior work, *random differential testing* and testing via *equivalence modulo inputs* (EMI), we present several strategies for random generation of deterministic, communicating OpenCL kernels, and an injection mechanism that allows EMI testing to be applied to kernels that otherwise exhibit little or no dynamically-dead code. We use these methods to conduct a large, controlled testing campaign with respect to 21 OpenCL (device, compiler) configurations, covering a range of CPU, GPU, accelerator, FPGA and emulator implementations. Our study provides independent validation of claims in prior work related to the effectiveness of random differential testing and

One of the principal challenges for an OpenCL implementer in achieving this portability guarantee for a given device is the construction of a correct compiler for OpenCL C, the C-like programming language in which kernel functions are written. To be conformant, the compiler must support the full range of OpenCL language constructs, which includes a three-layer memory hierarchy, a rich set of vector data types and operations, read-modify-write atomics, and barrier synchronization. To be practical, the compiler must perform aggressive, device-specific optimizations since performance is the *sole* reason for adoption of OpenCL. OpenCL compiler reliability is especially crucial because, by default, compilation is performed *online*. OpenCL-accelerated applications are written in a device-agnostic manner and the kernels used by an application are compiled at runtime by the drivers of available devices. Online

Current vendor testing strategies: inadequate

PLDI'15 paper (Imperial) reports more than **50 bugs** in commercial OpenCL implementations, from multiple **major vendors**

TTP Solution

Find compiler bugs automatically via **random differential testing**

CLsmith

Imperial College London

generate

Large set of feature-rich OpenCL test cases

test

based on

University of Utah

Collective Knowledge

dividiti

- Crowd-sourcing of results from diverse platforms
- Visualisation of test results
- Bug identification
- Bug ranking
- Test case reduction
- Bug reproduction

"Crowd-check" OpenCL platforms from major vendors

…and more

CK offers a **seamless, generic framework** for crowd-sourcing, managing and analysing **large data sets** arising from computer science experiments, including **many-core API fuzzing**

TTP Impact

A many-core crowd-testing service

CK/CLsmith will lead to a service for **rapid testing** of prototype API implementations

Increased value of vendor tools

Eliminating key defects early will lead to reliable vendor tools, with **higher value** to customers

Better many-core software

Reliable vendor tools will enable construction of **safer**, more **secure** many-core software for **society**

TTP Facts

Contact: Alastair F. Donaldson
E-mail: alastair.donaldson@imperial.ac.uk
TETRACOM contribution: 30,132 €
Duration: 01/02/2016-30/06/2016