Many-Core Compiler Fuzzing
Christopher Lidbury,1 Andrei Lascu,1 Nathan Chong,2 Alastair F. Donaldson1
1Imperial College London, 2University College London

Research aim: validate effectiveness of existing compiler testing methods…

Generating Random Kernels
Our CLsmith tool generates concurrent-yet-deterministic OpenCL kernels
Vector mode: exercises vector operations
uint4 v = clamp(a, b.xxyy, clz(c));
Barrier mode: orchestrates race-free shared memory communication via barriers
Atomic mode: exercises read-modify-write operations in a deterministic way
if(atomic_inc(c) == 42) {
  // Thread-insensitive code
}
Which thread enters: non-deterministic
Effect of section: deterministic

Dead-By-Construction Injection
Equip kernel with extra Boolean parameter, DEAD
Inject code conditional on DEAD

Vector mode: exercises vector operations
Barrier mode: orchestrates race-free shared memory communication via barriers
Atomic mode: exercises read-modify-write operations in a deterministic way

Tested 21 OpenCL (device, compiler) combinations; major vendors + open source
Results show that RDT and EMI are effective in this domain
More than 50 distinct bugs found, many now fixed by vendors

Test programs and full results: multicore.doc.ic.ac.uk/tools/CLsmith