

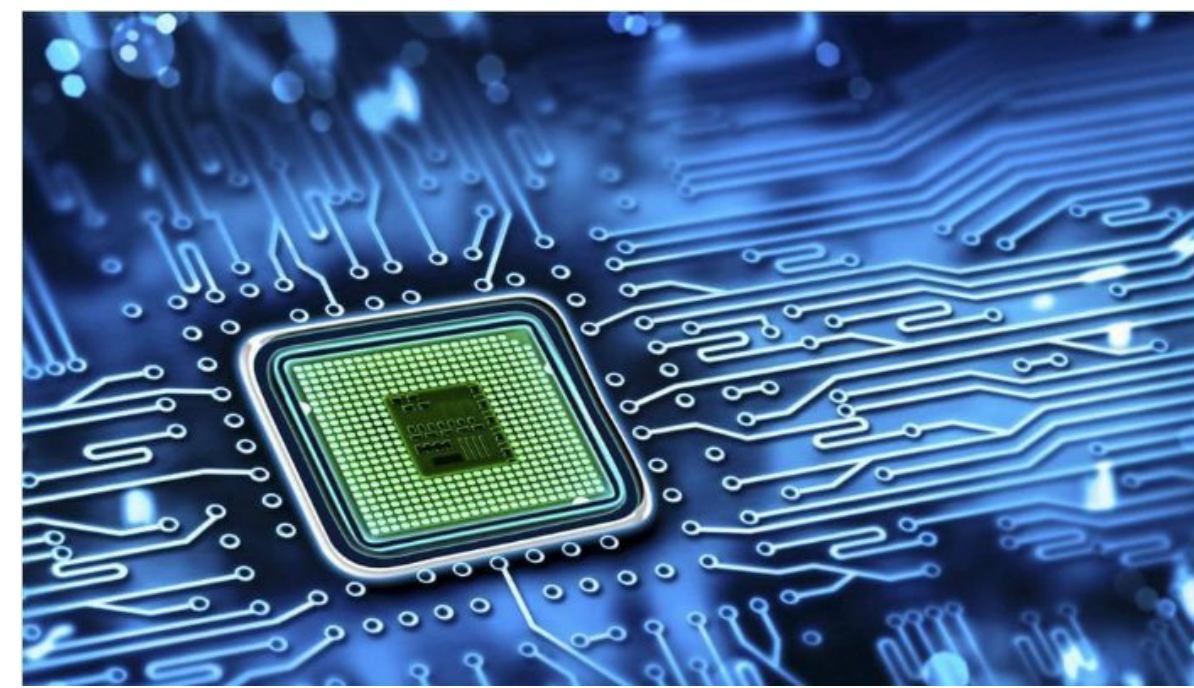
Compiler-Assisted Test Acceleration Using GPUs

Vanya Yaneva Ajitha Rajan Christophe Dubach

Motivation

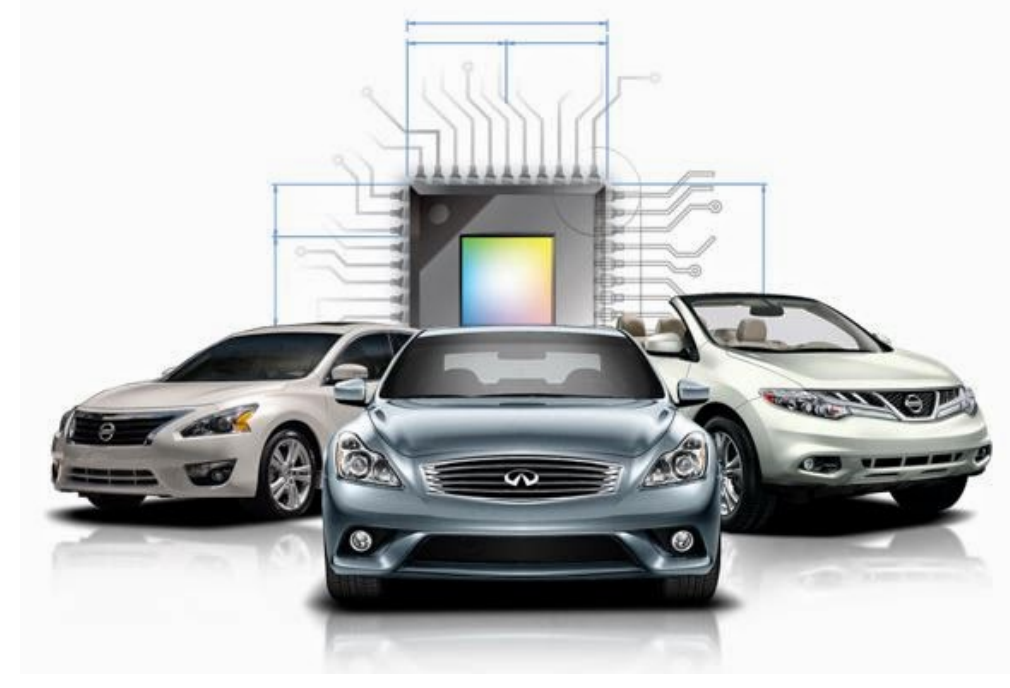
Background:

Software is ubiquitous and its correctness is critical.



Problem:

Functional software testing is crucial, but **extremely time consuming**.



Proposed solution:

Execute test cases in parallel on the GPU threads.



Benefits

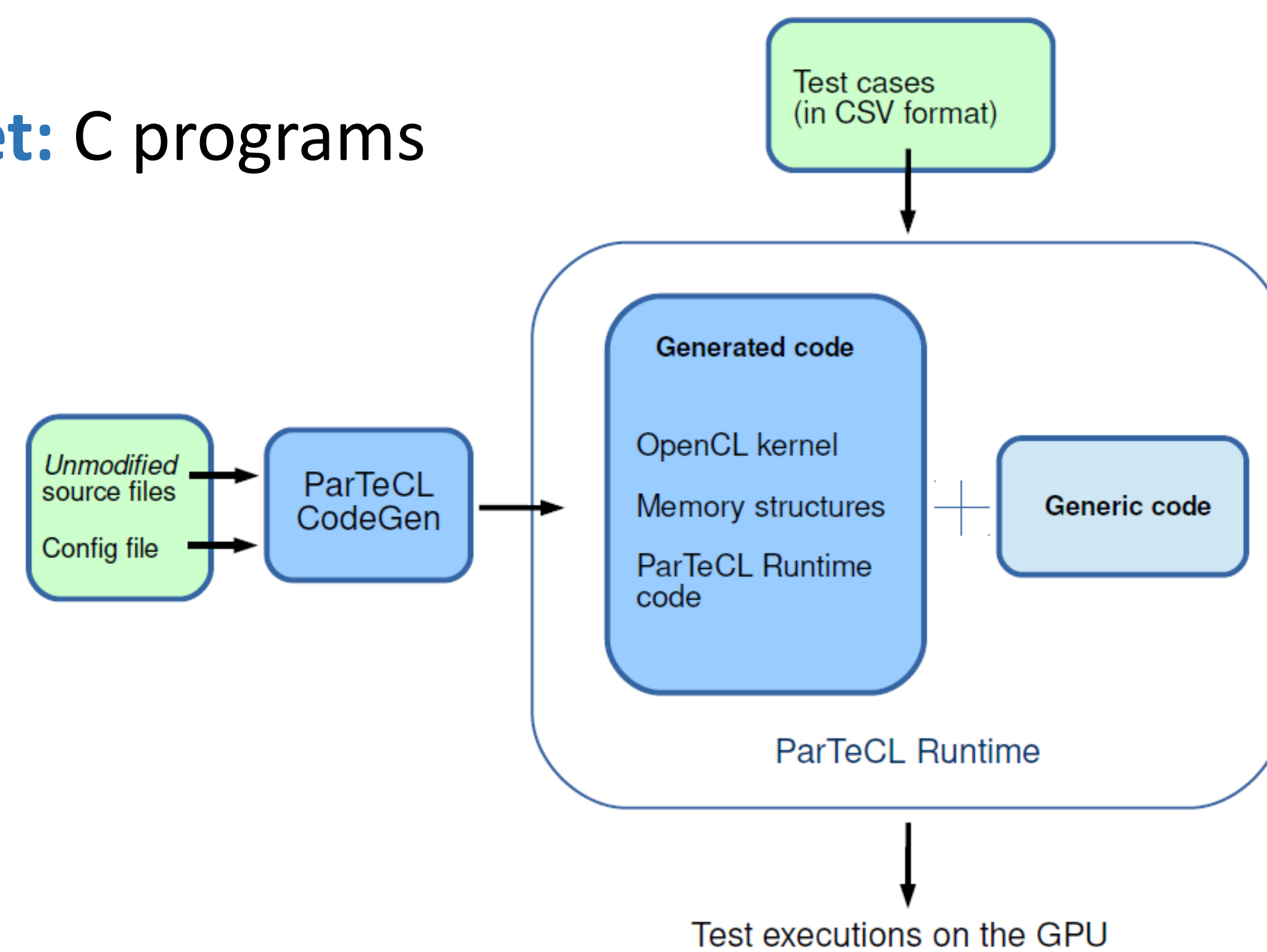
- **Reduced:**
 - testing time
 - cost of testing infrastructure & maintenance
 - energy consumption
- **Increased:**
 - Speed of testing
 - Frequency of testing
 - Fault finding

Approach

Implemented two systems to automate test execution on the GPU:

- **ParTeCL¹ CodeGen:** generates an OpenCL kernel from the source code of the tested program
- **ParTeCL Runtime:** launches the test cases on the GPU threads

Target: C programs



¹ParTeCL – Parallel Testing with OpenCL

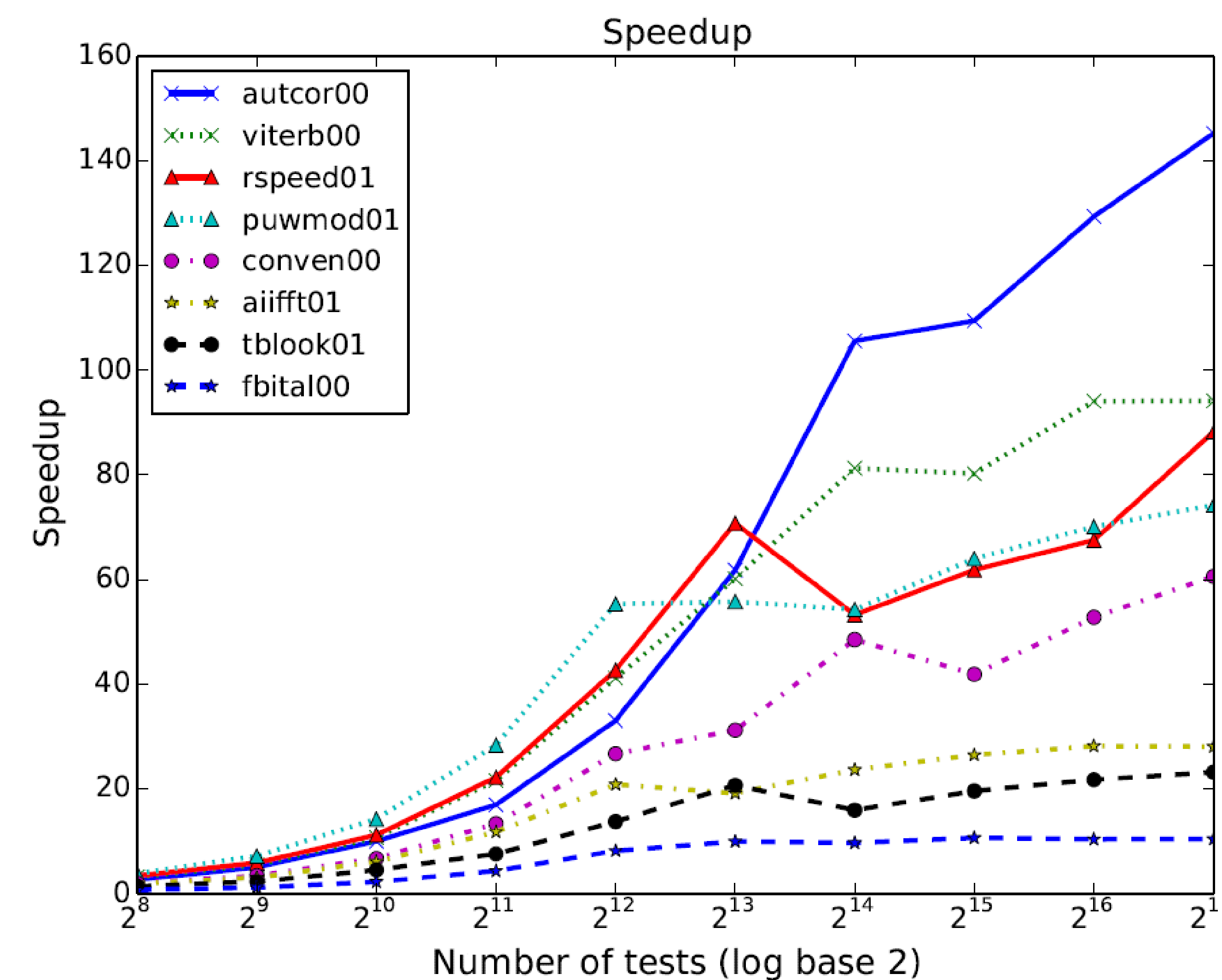
Results

Subjects: 8 programs from the **EEMBC** benchmark suite

Hardware: NVidia Tesla K40m

Results:

- **Average speedup of 33x** compared to a single CPU
- **Correctness verified:** testing results on the GPU are the same as on the CPU



Limitations & Future work

1. **Time to transfer data between the CPU and GPU adds an overhead:** Overlap data transfers with kernel execution.
2. **Control-flow divergence limits performance:** Group test cases based on control-flow paths.
3. **Not all C features are readily supported for compilation on the GPU:** Implement code transformations in ParTeCL Codegen to handle those.
4. **Perform empirical evaluation with a variety of benchmarks.**
5. **Compare with execution times on other architectures.**



THE UNIVERSITY of EDINBURGH
informatics

EPSRC Centre for Doctoral Training in
Pervasive Parallelism