Using Performance Correlation Heuristics To Accelerate Compiler Sequence Specialization on GPUs

Ricardo Nobre, Luís Reis, João M. P. Cardoso
University of Porto and INESC TEC, Porto, Portugal
{rjfn,luis.cubal}@fe.up.pt,jmpc@acm.org

Abstract

Iterative compilation relying on compiler phase ordering specialization (i.e., customizing the order of compiler passes for each program/function) has been shown to be able to improve the performance of code targeting GPUs. However, it can be too costly due to the overhead of compiling and testing hundreds/thousands of phase orders. An existing approach that has been tested on CPUs is to first generate a small set of complementary high-quality compiler phase orders during a training phase, and then, given new code, evaluate only these phase orders. In this paper, we evaluate the viability of this type of approach in the context of GPU kernel optimization. In addition, we propose and test a set of heuristics designed to further reduce the number of evaluated compiler phase orders, by taking into account the correlation between the speedups of compiler phase orders. We measured the impact of our heuristics using the PolyBench/GPU OpenCL benchmark suite on an NVIDIA GP104 (Pascal) GPU. The use of our heuristics allows to achieve a geometric mean speedup of 1.64×, using cross-validation, while only evaluating 3 non-standard compiler sequences, while not using heuristics required evaluating 5 non-standard compiler sequences to achieve the same GPU kernel performance. This represents 96.7% of the geometric mean improvement achieved in our experiments by iterative compilation with thousands of executions. We believe that these approaches and heuristics can increase the suitability of compiler phase ordering specialization to a wide range of compiler users concerned with GPU kernel execution performance.

Keywords phase ordering, GPGPU, LLVM, optimization, compilation

CPC 2018, ,
2018.