VSC School Project: Performance Enhancements of Algebraic Multigrid Methods in ViennaCL

Florian Rudolf[a], Andreas Morhammer[a], Karl Rupp[a], Josef Weinbub[b]

[a] *Institute for Microelectronics, TU Wien, Austria*
[b] *Christian Doppler Laboratory for HPTCAD, Institute for Microelectronics, TU Wien, Austria*

Algebraic multigrid (AMG) methods are attractive for the large-scale solution of sparse systems of equations because of their asymptotic optimality and their black-box nature. As a consequence, a lot of research has been conducted on how to parallelize AMG methods for supercomputers as well as for novel many-core architectures such as graphics processing units (GPUs).

The two main stages of an AMG method are the *setup* stage and the *solution* stage. In the setup stage the grid hierarchy is constructed from the entries of the sparse system matrix. The performance bottlenecks in the setup stage are smoothing operations (typically variants of sparse matrix-vector products) and sparse matrix-matrix products. In the solution stage, the grid hierarchy is traversed to compute defect corrections. The performance bottlenecks in the solution stage are smoothing operations and grid transfer operations, both typically performed via sparse matrix-vector products.

This VSC School project focuses on enhancing the GPU-accelerated implementations of AMG methods in the free open source library ViennaCL[1]. Our earlier work focused on enhancing sparse matrix-vector and sparse matrix-matrix products as standalone kernels [1,2]. In this work, we present results of our now highly-tuned AMG implementation consisting of the aforementioned highly-optimized kernels and new kernels for the required graph algorithms [3].

Results: We compared the total execution times obtained for an unsmoothed, aggregation-based AMG method as defined as our benchmark at the beginning of the VSC School project. Our results were obtained on a dual-socket Intel Xeon E5-2620 machine equipped with an NVIDIA Tesla K20m GPU and can be reproduced by running the publicly available benchmark code on GitHub[2]. Fig. 1 shows that the AMG implementation in the latest ViennaCL 1.7.1 release is by a factor of four to six faster than the AMG implementation in ViennaCL 1.6.2 available at the start of the VSC school. The saturation of execution times at system sizes below 10^4 unknowns for the new implementation stems from PCI-Express latencies.

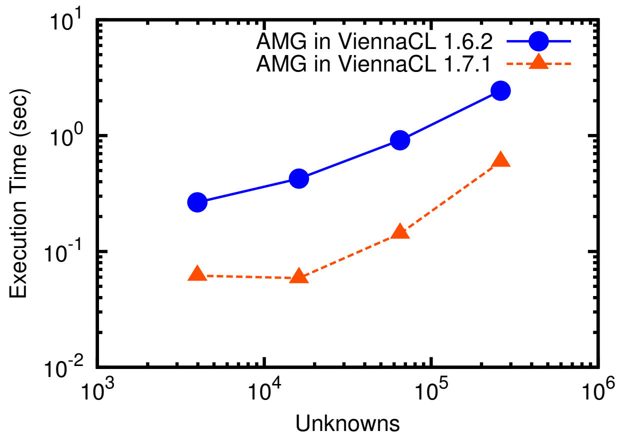

Fig. 1: Execution times for a full AMG solver at different system sizes on an NVIDIA Tesla K20m.

Financial support by the VSC Research Center, the Austrian Science Fund (FWF), the Austrian Federal Ministry of Science, Research and Economy, and the National Foundation for Research, Technology and Development is gratefully acknowledged.

References

[1] Rupp, K., Tillet, Ph., Rudolf, F., Weinbub, J., Morhammer, A., Grasser, T., Jüngel, A., and Selberherr, S., SIAM J. Sci. Comp., **38**, S412 (2016).

[2] Morhammer, A., Rupp, K., Rudolf, F., Weinbub, J., AHPC 2016, 23 (2016).

[3] Bell, N., Dalton, S., and Olson, L., SIAM J. Sci. Comp., **34**, C123 (2012).

[1] http://viennacl.sourceforge.net/
[2] https://github.com/viennacl/vsc-school/