VSC School Project:

Optimized sparse matrix-matrix multiplication for multi-core CPUs, GPUs, and MICs

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

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

Sparse matrices are extensively used in areas such as linear algebra, data mining, or graph analytics. One of the fundamental operations is general sparse matrix-matrix multiplication (SpGEMM), where our primary interest is in computing coarse grid operators in algebraic multigrid methods [1]. While certain applications provide additional information to derive optimized sparse matrix-matrix multiplications, a fast and general SpGEMM is desirable from an abstraction point of view. As a consequence, parallel implementations of SpGEMM are provided by several libraries including the Math Kernel Library (MKL) by INTEL for CPUs and MICs, and CUSP as well as CUSPARSE by NVIDIA for NVIDIA GPUs.

In this work we present optimization results for SpGEMM on shared memory systems equipped with multi-core CPUs, GPUs, or MICs. We build on top of previous work on optimizing SpGEMM for NVIDIA GPUs [2], generalize the optimization techniques to other architectures [3], and derive fast implementations for hardware from all major vendors: First, an SpGEMM kernel implementation based on second-generation advanced vector extensions (AVX2) intrinsics merging multiple rows concurrently on the latest Haswell Xeon CPU line, and an implementation based on 512-bit wide AVX intrinsics on Xeon Phi (KNC) is discussed. Second, an embedded performance model for estimating the work required by each thread is introduced, resulting in improved load balance across threads. Third, our contribution for a GPU-based SpGEMM is a refinement of the recently proposed row-merging algorithm proposed in Ref. [2] by reducing the memory footprint and the number of kernel launches. While the original row-merging algorithm has memory overheads of at least the size of the result matrix, the additional memory required by our algorithm depends only on the total number of threads and the maximum number of nonzeros in the right hand side factor matrix. A comparison with MKL, CUSP, and CUSPARSE in Figure 1 demonstrates a 50 percent performance gain over INTEL's MKL library on a recent Haswell-based Xeon system on average. A two-fold performance gain over CUSP and CUSPARSE is demonstrated on an NVIDIA Tesla K20m. Also, we present the first implementation of an efficient SpGEMM on AMD GPUs based on row-merging.

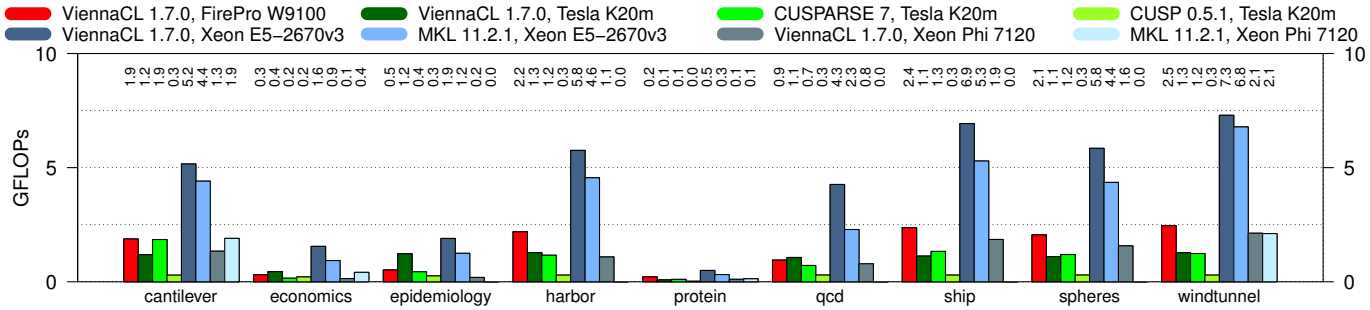

Fig. 1: Performance comparison of sparse matrix-matrix multiplication routines.

References

[1] Trottenberg, U., Oosterlee, C. W., and Schüller, A., Multigrid, Academic Press (2001).

[2] Gremse, F., Höfter, Schwen, L. O., Kiessling, F., and Naumann, U., GPU-Accelerated Sparse Matrix-Matrix Multiplication by Iterative Row Merging. SIAM J. on Sci. Comp., **37**(1):C54 (2015).

[3] Rupp, K., Tillet, Ph., Rudolf, F., Weinbub, J., Morhammer, A., Grasser, T., Jüngel, A., Selberherr, S., ViennaCL – Linear Algebra Library for Multi- and Many-Core Architectures, submitted to SIAM J. Sci. Comp.