Matrix-free multi-GPU Implementation of Elliptic Solvers for strongly anisotropic PDEs

Dr. Eike Mueller, **Prof. Rob Scheichl**, Dept. of Mathematical Sciences, University of Bath,
Prof. Eero Vainikko, Faculty of Mathematics and Computer Science, University of Tartu

Part of this work was carried out in collaboration with Ms. Xu Guo, Mr. Sinan Shi (EPCC) and Dr. Benson Muite (KAUST).

October 30, 2013

Manycore chip architectures such as Graphics Processing Units (GPUs) have been shown to be highly efficient both in terms of total performance and small power consumption in many areas of Scientific Computing. However, implementing memory-bound applications such as elliptic solvers for large partial differential equations (PDEs) remains a challenge.

We consider iterative solvers for strongly anisotropic PDEs which arise in many problems in geophysical modelling in "flat" domains, such as numerical weather- and climate prediction (NWP), global ocean models or subsurface reservoir- and flow- simulations. For example, an equation for the pressure correction has to be solved in a thin spherial shell representing the earth's atmosphere at every timestep in many NWP codes. To improve the accuracy of the model, the spatial resolution is continuously increased and this leads to significant computational challenges as forecasts have to be delivered on operational timescales.

Optimised "matrix-free" GPU implementation. We developed an optimised GPU implementation of a preconditioned conjugate gradient (PCG) algorithm which deals with the strong vertical anisotropy by relaxing all degrees of freedom in a vertical column simultaneously [1]. The main components of the algorithm are a sparse matrix-vector product (SpMV) and a tridiagonal solve based on the Thomas algorithm (see e.g. [2]). To achieve coalesced memory access in the tridiagonal solver, it is crucial to store field vectors such that the horizontal index is contiguous in memory. We demonstrated that because of the tensor product structure of the grid and the operator the performance can be improved significantly by using a "matrix-free" implementation where the local matrix stencil is recalculated on-the-fly instead of reading it from memory. This is illustrated in Fig. 1 (left) where we compare the implementation based on the CuSparse library to our optimised code. In addition, further speedups could be achieved by reducing global memory access through fusing different GPU kernels. In total our optimised algorithm can harness $25\% - 50\%$ of the theoretical peak global memory bandwidth on an nVidia Fermi M2090 GPU and is $50\times$ faster than the sequential CPU implementation on a

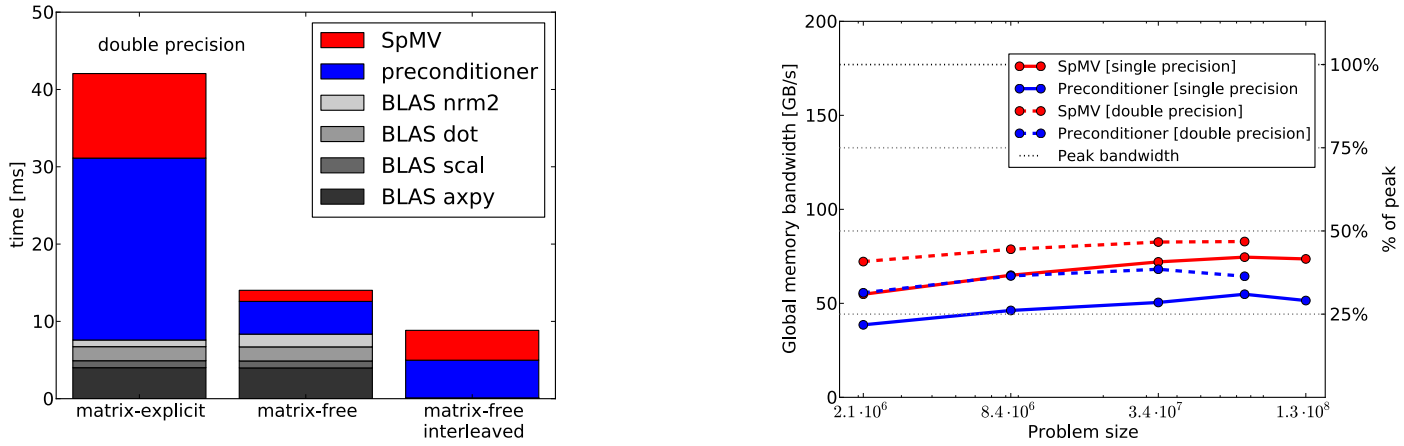

Figure 1: Time per iteration for different GPU implementations of the PCG solver (left) and global memory bandwidth for different problem sizes (right). The computationally most expensive kernels are highlighted in red and blue.

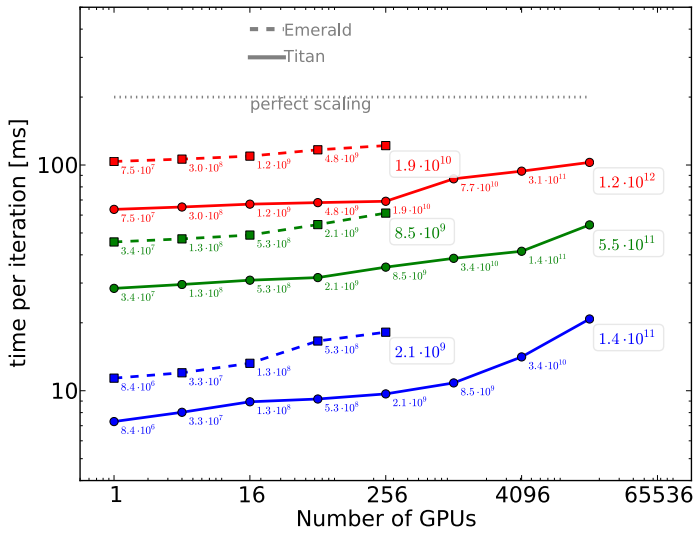

Figure 2: Weak scaling of the time per iteration on Emerald (dashed) and Titan (solid lines). Figure provided by Benson Muite.

Sandybridge chip (see Fig. 1, right).

Multi-GPU solvers. To reduce the global model resolution to the order of a few kilometers, an algebraic system with more than 10 billion unknowns needs to be solved, which is not possible on a single GPU. Using the Generic Communication Library (GCL) [3] we extended our approach to GPU clusters and demonstrated its scalability on EMERALD and on up to 16384 GPUs on the Titan supercomputer (Fig. 2). The largest system we solved had more than 1 trillion unknowns and required a time per iteration of around 100 milliseconds.

Recently we have shown [4] that on CPU clusters bespoke geometric multigrid solvers can reduce the number of iterations significantly. Compared to Krylov-subspace solvers the only additional ingredients are intergrid operators, which can partially be fused with the existing kernels to improve cache resuage. We are currently working on the GPU implementation of the algorithm and hope to report some preliminary results soon.

References

[1] Eike Mueller, Xu Guo, Robert Scheichl, and Sinan Shi. Matrix-free GPU implementation of a preconditioned conjugate gradient solver for anisotropic elliptic PDEs. *accepted for publication in Computing and Visualization in Science*, 2013. preprint: http://arxiv.org/abs/1302.7193.

[2] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing, 3rd Edition*. Cambridge University Press, New York, 2007.

[3] Bianco, Mauro (CSCS) et al. The Generic Communication Library. unpublished.

[4] Eike Mueller and Robert Scheichl. Massively parallel solvers for elliptic pdes in numerical weather- and climate prediction. *submitted to Quarterly Journal of the Royal Meteorological Society*, 2013. preprint: http://arxiv.org/abs/1307.2036.