Delft University of Technology

Project: Iterative Sparse Solvers on the SX Aurora Vector Engine

Supervisor(s):
Jonas Thies
Martin van Gijzen

June 4, 2021

Project Description

At the core of many scientific and engineering simulations one has to efficiently solve a sparse matrix problem. Most commonly, linear systems of equations have to be solved (for instance after discretizing a partial differential equation using the finite element method). Sparse eigenvalue problems arise, e.g., in quantum mechanics, where the key equation to be solved is typically a Schrödinger equation.

From the perspective of high performance computing (HPC), sparse matrix algorithms lead to memory-bound behavior: the factor determining the execution time of a well-optimized code is the amount of data that has to be moved around before or after being processed. Vector processors have been developed to address this bottleneck since the dawn of HPC by providing an exceptionally high memory bandwidth. In this thesis we will explore data structures and code optimization for one of the fastest HPC processors available, the SX Aurora TSUBASA designed by the Japanese company NEC. With a total memory bandwidth of 4.8 TB/s it matches the speed of a GPU system but with only 32 powerful vector cores, which makes it easier to program and optimize your code.

The thesis will focus on one or two key iterative methods in numerical linear algebra, such as a Krylov subspace method or Jacobi-Davidson (to be discussed). An existing software framework for implementing iterative methods in a hardware-transparent way will be used on the algorithm side [2], see https://bitbucket.org/essex/phist/). In order to perform the required computational tasks (matrix and vector operations) on the accelerator cards, a number of basic operations must be implemented separately. This will constitute the main programming task of the thesis as phist already includes a number of iterative methods. These basic operations can be implemented e.g. in C, Fortran or C++, with OpenMP and MPI for parallelization on a single card and between the four cards available on the system, respectively.

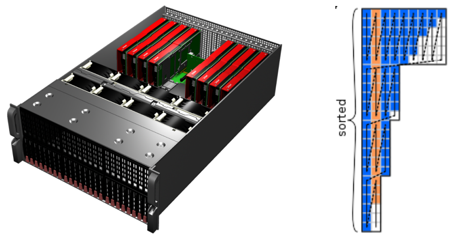

Figure 1: Left: a glimpse at the SX Aurora. Right: Sliced ELLpack storage (SELL) for vectorized sparse matrix computations [1].

What you will do

- Get to know the SX Aurora by going through an NEC performance optimization tutorial.
- Get to know the rudimentary linear algebra library ghostxx (a C++ code in an early state of development)
- Implement a suitable sparse matrix data structure, e.g. the sliced ELLpack format (see right figure).
- Benchmark complete algorithms with sparse matrix problems from actual applications using existing solvers in phist.
- Depending on your interests:
  - implement an iterative solver of your own in phist.
  - more detailed performance analysis and optimization using tools and performance models

What we offer

- you will get a chance to collaborate with the German Aerospace Center's institute for Software Technology (DLR-SC) which hosts the Aurora system,
- a trip to Cologne is planned for a one week 'Aurora hackathon' where you can meet HPC experts from the DLR
- two friendly, relatable and helpful supervisors

Contact

If you are interested in this project and/or have further questions please contact Jonas Thies, j.thies@tudelft.nl, and Martin van Gijzen, m.b.vangijzen@tudelft.nl.

References

[1] M. Kreutzer et al. A unified sparse matrix data format for efficient general sparse matrix-vector multiplication on modern processors with wide SIMD units. *SIAM J. Sci. Comp.*, 36(5):C401–C423, 2014. doi: http://dx.doi.org/10.1137/130930352. URL http://epubs.siam.org/doi/abs/10.1137/130930352.

[2] J. Thies et al. PHIST: a Pipelined, Hybrid-parallel Iterative Solver Toolkit. *ACM Trans. Math. Software*, 6, 2020. URL https://dl.acm.org/doi/abs/10.1145/3402227.