Parallel Autotuning Compilation for Algorithmic Modelica on Nvidia 2 teraflop 2050 GPU

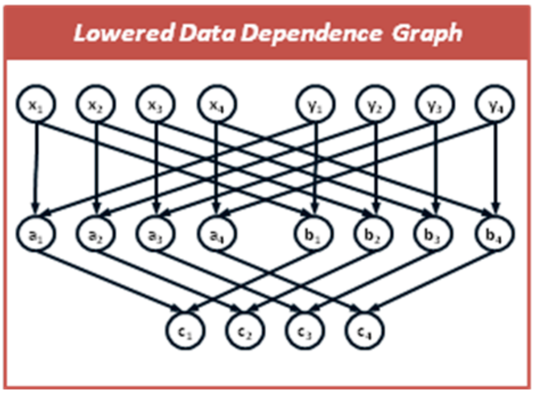
Contact: Peter Fritzson (peter.fritzson@liu.se, tel: 0708-281484)
or Mahder Gebremedhin (Mahder.Gebremedhin@liu.se),
or Olena Rogovchenko (olena.rogovchenko@liu.se)
PELAB – Programming Environment Lab, Institutionen för Datavetenskap
www.openmodelica.org

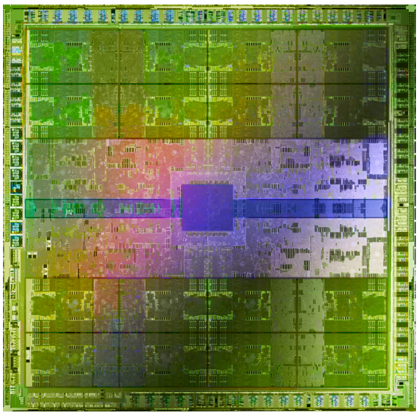
At PELAB, together with the Open Source Modelica Consortium (an international open source effort supported by 38 organizations, see www.openmodelica.org) the OpenModelica environment including the OpenModelica Compiler (OMC) of the Modelica language including MetaModelica extensions is developed. Modelica is a high level language supporting equations and matrix operation. The development is open source.

Currently OMC compiles Modelica/MetaModelica into C-code via several optimizing steps. The development is supported by an Eclipse plug-in MDT (Modelica Development Tooling), also including a debugger, and a template language already used for developing code generators to C and C#. There has earlier been developed several parallel code generator prototypes ([2] – [5]) in the OpenModelica system including generation of OpenCL code for Nvidia. However, static compilation schemes sometimes fail to parallelize due to conservative assumptions of dependencies.

The goal of this master thesis project is to design and implement an efficient parallel autotuning compilation scheme from Modelica to the NVIDIA Fermi GPGPU. The development need not be done from scratch, it can build on and re-use part of 7000 line C++ implementation with good results for the Cell BE described in [1]. The idea is run-time construction of intermediate code for fragments of models/programs with many matrix/array operations. The run-time construction gives exact knowledge of dependencies. The intermediate code is downloaded into the GPU and executed whenever some computation results is needed. The master thesis work includes adapting this approach to the Modelica language, and to Fermi GPGPU instead of the previous Cell BE.

The master thesis project requires knowledge of compiler construction, parallel programming, as well as some experience and interest in advanced programming.





References:

[1] Raymes Khoury, Bernd Burgstaller and Bernhard Scholz. Accelerating the Execution of Matrix Languages on the Cell Broadband Engine Architecture. Accepted to the IEEE Transactions on Parallel and Distributed Systems, 2012. (Send email to Mahder or Olena to get a .pdf copy of the paper)

[2] Mahder Gebremedhin. ParModelica: Extending the Algorithmic Subset of Modelica with Explicit Parallel Language Constructs for Multi-core Simulation. Master thesis 2011.
http://liu.diva-portal.org/smash/record.jsf?searchId=1&pid=diva2:451473.

[3] Afshin Hemmati Moghadam, "Modelica PARallel benchmark suite (MPAR) - a test suite for evaluating the performance of parallel simulations of Modelica models," Linköping University, Linköping, Sweden, Master Thesis LIU-IDA/LITH-EX-A— 11/042—SE, 2011.

[4] Kristian Stavåker. Contributions to Parallel Simulation of Equation-Based Models on Graphics Processing Units, Licentiate Thesis, Dec 2011. http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-71270

[5] Peter Aronsson. Automatic Parallelization of Equation-Based Simulation Programs. PhD thesis 2006. http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-7446.