CUDA Code Generation and Runtime Support for Parallel Extensions of the OpenModelica Compiler.

Target architecture: NVIDIA Tesla M2050 GPU (Fermi): 2 teraflop GPU

Contact: Peter Fritzson (peter.fritzson@liu.se, tel: 0708-281484)
or Mahder Gebremedhin (mahder.gebremedhin@liu.se, tel: 0760831822)
PELAB – Programming Environment Lab, Institutionen för Datavetenskap
www.openmodelica.org

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. The development is open source under the OSMC-PL and GNU V3 licenses.

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#.

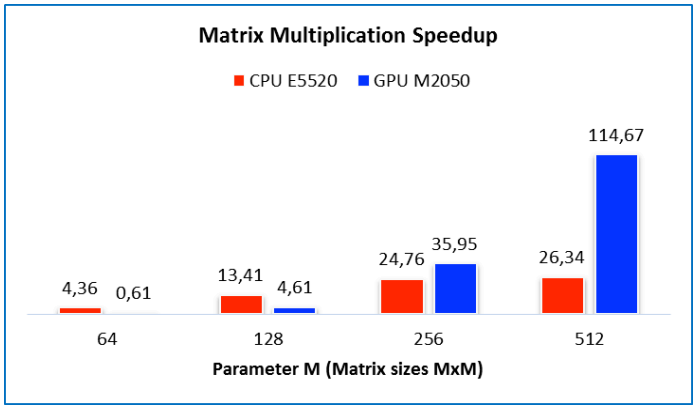
There has earlier been developed several parallel code generator prototypes for the OpenModelica compiler. Some of these are related to automatic parallelization and code generation while others focused on explicit parallelization and code generation.

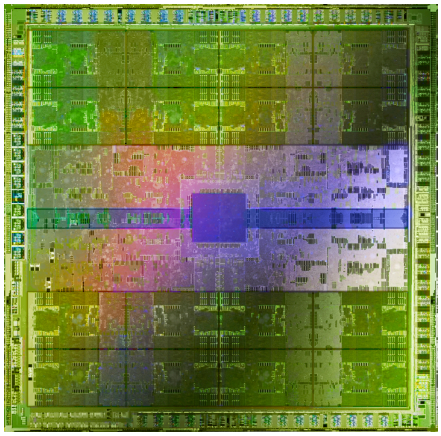
The most recent work was resulted in OpenCL/CUDA style explicit parallel programming extensions for the OpenModelica compiler to achieve data parallelism on modern GPGPUs. For example a new parallel for loop that distributes its computations across GPU processors and cores has been implemented. In addition support has been added for the different memory locations of the OpenCL memory model (data-parallel variables with different characteristics), OpenCL kernel functions and OpenCL parallel functions. This implementation generates OpenCL code for the parallel extensions.

The goal of this master thesis project is to improve the currently available extensions and to add CUDA code generation and runtime support for these new explicit parallel programming constructs in order to achieve speedups on NVIDIA GPUs. For example the parallel for loop mentioned earlier should be adapted to operate with CUDA, CUDA kernel functions should be supported etc.

If there is time, support for NVIDA's BLAS (Basic Linear Algebra Subprograms) implementation, CUBLAS will be added to the OpenModelica compiler as an option to the currently available sequential FORTRAN LAPACK interface.

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





References:

[1] 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.

[2] 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..

[3] Per Östlund. Simulation of Modelica Models on the CUDA Architecture. Master thesis 2010. http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-52060 .

[4] Hans Zima and Barbara Chapman. Supercompilers for parallel and vector computers . http://portal.acm.org/citation.cfm?id=89627.

[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.