Using "ArrayOL" to Identify Potentially Shareable Data in Thread Work-Groups of GPUs

{Wendell.Rodrigues, Frederic.Guyomarch, Jean-Luc.Dekeyser}@inria.fr

Université Lille1 Sciences et Technologies — lifl — INRIA

Contribution

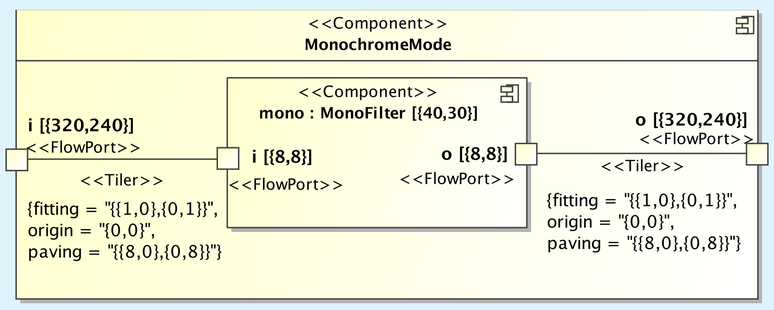
We propose an optimization method suitable to environments based on MDE and ArrayOL for code generation to GPU architectures. This method takes into account information from the dataflow modeling and data reuse by thread work-groups. The aim is to achieve better performances on data access in order to provide speed-up in parallel applications.

Background

OpenCL Automatic Code Generation

Gaspard2 is a framework based on Model Driven Engineering (MDE) and MARTE. This framework allows us generating code to several environments including massively parallel architectures such as GPU. Recent changes in Gaspard2 bring us a transformation chain (MARTE to OpenCL) able to create automatically code for OpenCL API in hybrid architectures. However, optimization levels currently detected by programming experts are challengers in MDE compilers. This approach adds the detection ability to identify potentially shareable data.

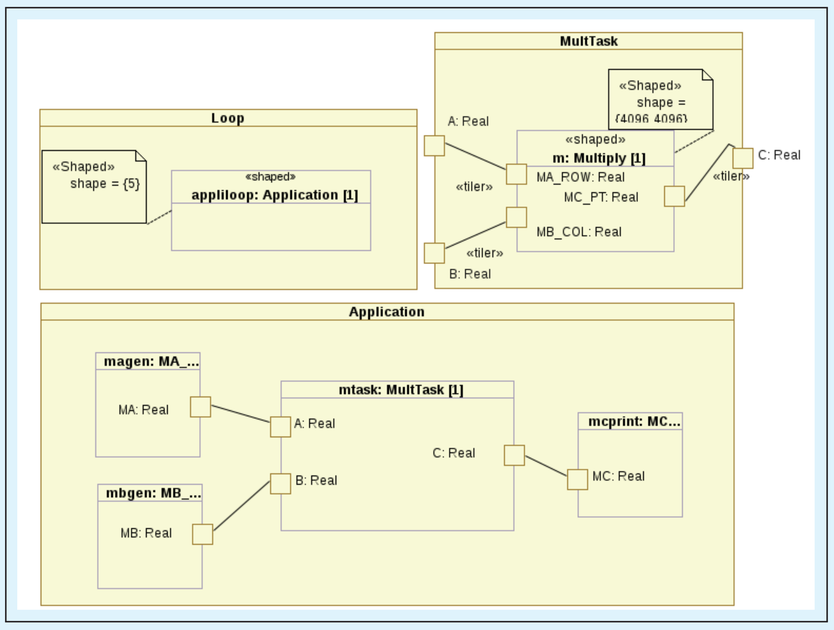
ArrayOL and Tilers



Tilers are stereotyped in connectors between ports around the application model. The model contains connectors with the stereotype *Tiler* linking a part A of shape M from a port of shape N to a port of shape P of a containing component. This topology means that each element of the pattern N at the iteration M is transmitted to the element the array P according to *origin*, *paving* and *fitting* attributes of the *Tiler* following this formula:

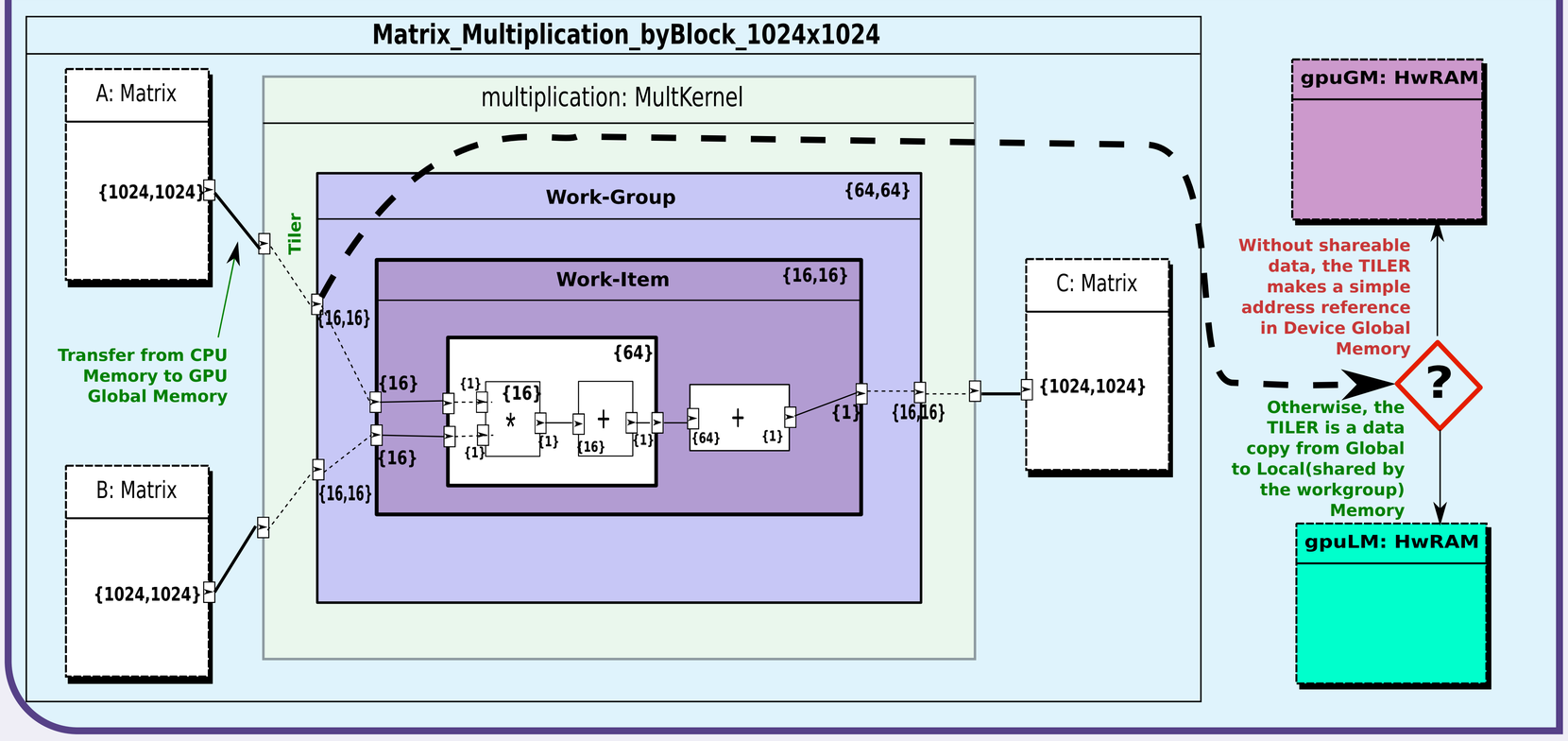
$$\{origin + paving.i + (fitting.j \mod shape) \mid 0 \leq i < M | 0 \leq j < N\}$$

Matrix Multiplication Model



- $MC = MA * MB$, one thread takes one line from MA and one column from MB and computes one element in MC;
- modeled using Papyrus UML Tool;
- *magen, mbgen and mcprint* are allocated onto the CPU processor;
- the repetitive task m is allocated onto the GPU processor;
- usually *ports* (variables) from GPU tasks are placed into the **device global memory**;

Refactored Model



Method

$\prod_{k=0}^{dim} S_{wi}[k]$: internal product of the shape of the repetion space of the work-item;

$\prod_{k=0}^{dim} S_{array}[k]$: internal product of the shape of the external array;

$\prod_{k=0}^{dim} S_{pattern}[k]$: internal product of the shape of the internal array (pattern):

Detecting Data Reuse: *if* $\frac{\prod_{k=0}^{dim} S_{wi}[k] * \prod_{k=0}^{dim} S_{pattern}[k]}{\prod_{k=0}^{dim} S_{array}[k]} > 1$ *then* ***transfer****=true*

Data Transfer between Global and Local Memories: $k = \lfloor \frac{\prod_{k=0}^{dim} S_{array}[k]}{\prod_{k=0}^{dim} S_{wi}[k]} + 0.5 \rfloor$

k: how many array elements are copied by a work-item.

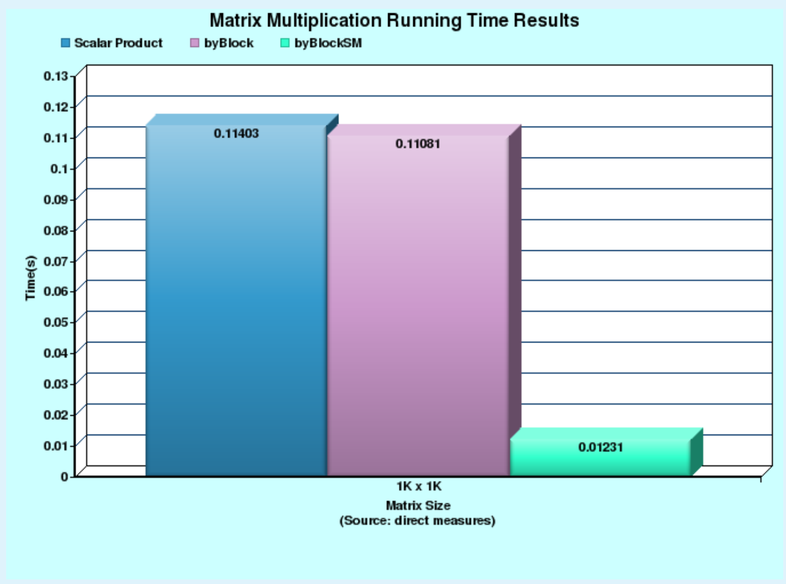
Open Questions

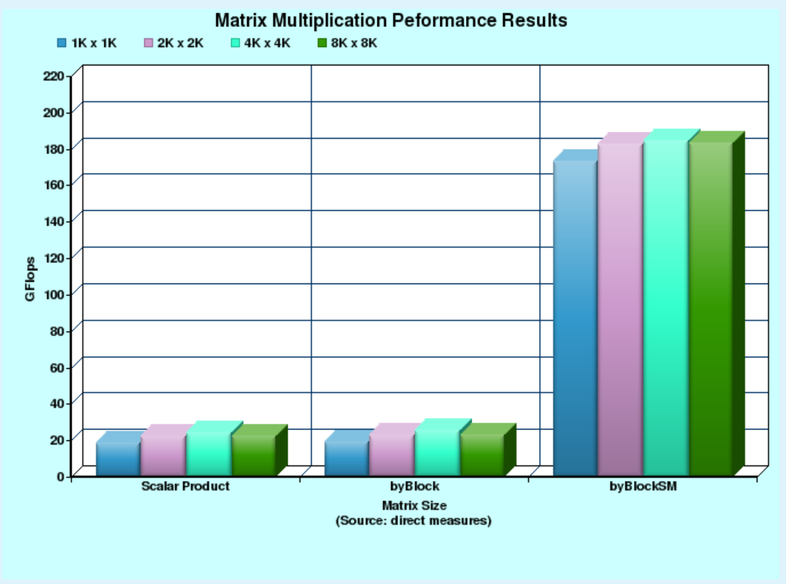
◎ How to create a rafactoring method suitable to hybrid environments.

◎ Benchmark for other application types and cases.

◎ Analysis of reading and writing ranking.

Results and Conclusions





⇒from a UML/MARTE high abstract model we can detect optimization levels in memory;
⇒these optimization additions can provide high gains in performance (9x in this case).

References

A. Wendell O. Rodrigues, Frédéric Guyomarc'h, and Jean-Luc Dekeyser. A*n MDE Approach for Automatic Code Generation from MARTE to OpenCL*. Technical report, INRIA Lille - RR-7525.

A. Gamatié, S. Le Beux, E. Piel, R. Ben Atitallah, A. Etien, P. Marquet, and J-L. Dekeyser. *A model driven design framework for massively parallel embedded systems*. ACM Transactions on Embedded Computing Systems (TECS). (to appear), 2011.

Funding

This work is supported by Nord-Pas de Calais Region, Valeo and GPUTech.