Compiling Image Processing Applications for Many-Core Accelerators
Pierre Guillou

To cite this version:
Pierre Guillou. Compiling Image Processing Applications for Many-Core Accelerators. ACACES Summer School: Eleventh International Summer School on Advanced Computer Architecture and Compilation for High-Performance and Embedded Systems, Jul 2015, Fiuggi, Italy. hal-01254412

HAL Id: hal-01254412
https://hal-mines-paristech.archives-ouvertes.fr/hal-01254412
Submitted on 12 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Compiling Image Processing Applications for Many-Core Accelerators

Pierre Guillou – CRI MINES ParisTech, PSL Research University

Image Processing

- Mathematical Morphology Base Operators
  - Arithmetic operators
    - Percentage (pixel [1] parameter, 1 input image)
    - Binary (pixel [2] parameter, 2 input images)
  - Morphological operators
  - Reducers
    - Neighbour selection: min/max/avg
  - Other operators
    - Global max/min/sum
  - Other operators
    - Threshold, mask, log2, ...

Mathematical Morphology: Image Analysis Theory and Technique Based

- Other operators
  - Reduction operators
  - Morphological operators
  - Arithmetic operators
  - Aggregation
  - Unrolling of converging loops

- Further optimisations
  - Streaming kernels
  - NoC Interface
  - Call graph optimisations
  - Operator library
  - Compute clusters
  - Debloking
  - Retina
  - Antibio
  - Burner

GMEAN

- Other programming models
  - Pthreads/OpenMP on compute clusters, communication library between clusters
  - OpenCL via local memory pagination

Example: Licence Plate Extraction

Sigma-C, a Dataflow Programming Language

Optimisations

- Unrolling of converging loops
- Arithmetic operators aggregation
- Generation of kernel-specific computations
- Data parallelism for compute-intensive operators

Compilation Chain

- Source-to-source compiler
- Call graph optimisations
- Target-specific compiler
- Compute kernels

Runtime Environment

- Host runtime on I/O clusters
- Accelerator runtime on compute clusters
- Control code
- Host runtime
- Accelerator runtime
- Compute clusters
- Transfer
- Compute clusters
- Transfer
- Compute clusters
- Compute clusters
- Host runtime
- Accelerator runtime
- Transfer
- Compute clusters
- Transfer
- Compute clusters
- Control code

Results: Execution Times and Energy Consumption (MPPA-256 = 1, lower is better)

Future Work

- Other programming models
  - Pthreads/OpenMP on compute clusters, communication library between clusters
  - OpenCL via local memory pagination
- Improve data-parallelism to take better advantage of the current architecture
- Implement more complex algorithms: watershed, arrow, labelling, minimal...

References

Pierre Guillou, Fabien Coelho, and François Irigoin.
Automatic Streamization of Image Processing Applications.
The 27th International Workshop on Languages and Compilers for Parallel Computing (LCPC), 2014.