Compiling Image Processing Applications for Many-Core Accelerators
Pierre Guillou

To cite this version:

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.
**Image Processing**

- **Streamline kernels**
- **NoC Interface**
- **Call graph optimisations**
- **Operator library**
- **Compute clusters**
- **Deblock**
- **OOP**
- **Retina burner**
- **Deblock**
- **Antibio**
- **GMEAN**
- **LP**
- **Toggle**
- **Antibio**
- **GMEAN**
- **LP**

**Mathematical Morphology Base Operators**

- Arithmetic operators
  - Addition, subtraction, multiplication, division
  - Comparison operators: greater than, less than, equal
- Morphological operators
  - Erosion, dilation, opening, closing
- Reduction operators
  - Mean, median, mode
- Other operators
  - Threshold, mask, erosion, dilation

**Implements more complex algorithms:** watershed, arrow, labelling, minima, ...

**Improve data-parallelism to take better advantage of the current architecture**

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

**Compilation Chain**

- Original application
- Source-to-source compiler
- Call graph optimisations
- Library
- Target-specific compiler
- Compute libraries

**Runtime Environment**

- Control code
- Host runtime
- Accelerator runtime on I/O clusters
  - I/O cluster
  - Compute clusters
  - System core
  - Compute core
  - Local memory
  - Shared memory (2 MB)
  - DDR (3 GB)

**Optimisations**

- Unrolling of converging loops
- Arithmetic operators aggregation
- Generation of kernel-specific convolutions
- Data parallelization for compute-intensive operators

**Future Work**

- Other programming models:
  - OpenCL, OpenMP on compute clusters, communication library between clusters.
  - OpenCL via local memory pagination.
- Improve data-parallelism to take better advantage of the current architecture.

**References**

Pierre Guilou, Fabien Coulo, and François Irigoïn.
Automatic Streamlining of Image Processing Applications.
The 27th International Workshop on Languages and Compilers for Parallel Computing (LCPC), 2014.