Compilation d'applications de traitement du signal sur accélérateurs matériels à haute efficacité énergétique

Rapport d'avancement 1re année

Pierre Guillou

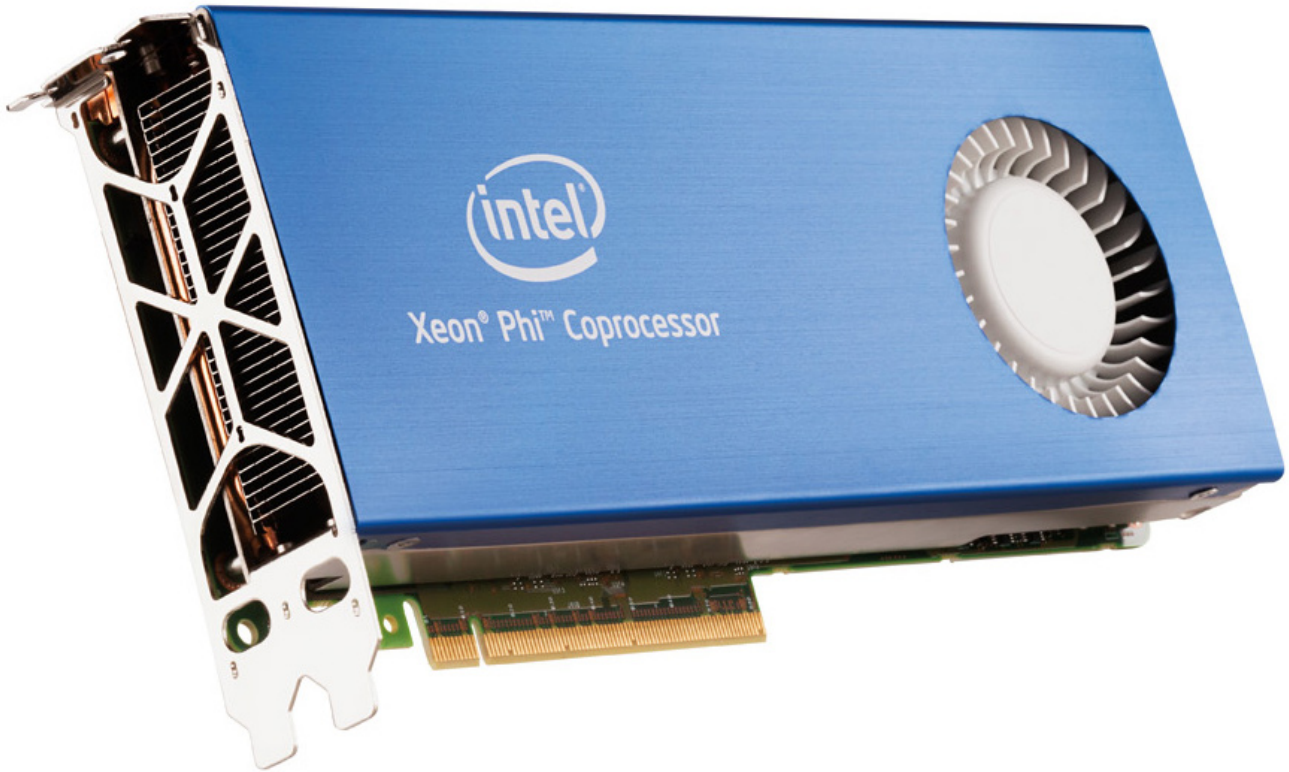
Directeur de thèse : François Irigoin
Maître de thèse : Fabien Coelho
MINES ParisTech – Centre de recherche en informatique

26 mai 2014

Des architectures matérielles variées

Tianhe-2, le superordinateur le plus puissant au monde

Des architectures matérielles variées



Le coprocesseur Intel Xeon Phi, avec 61 cœurs de calcul

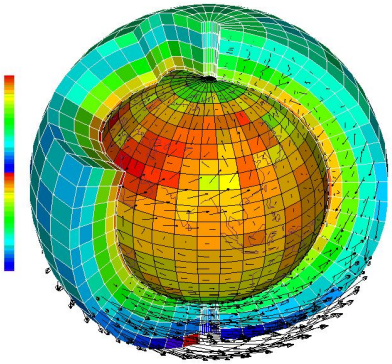
Des architectures matérielles variées

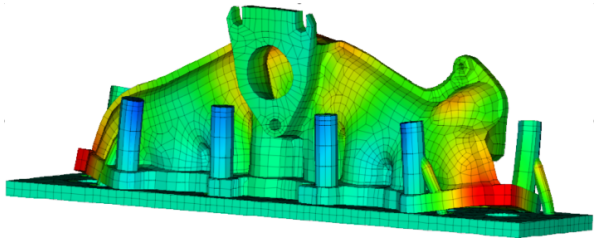
Un iPhone 5S

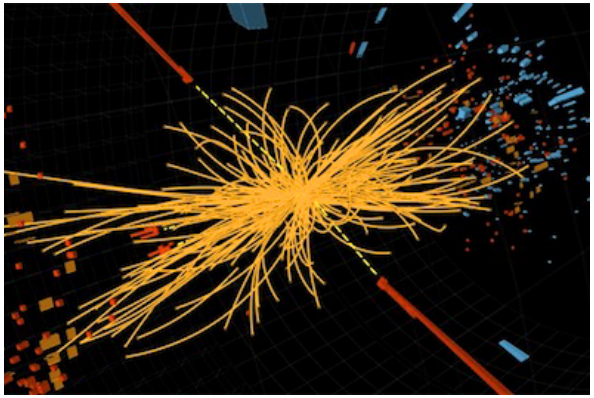
Des architectures matérielles variées

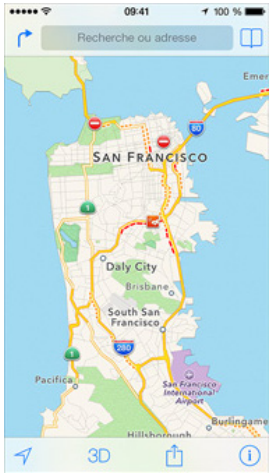
La console portable NVIDIA Shield

Des applications poussées









Entre les deux...

Coûts de développement

- compétences spécifiques dans l'architecture cible
- portabilité (plus petit dénominateur commun) vs performance

Consommation d'énergie

- facteur critique
  - HPC, datacenters
  - embarqué
- nouvelles architectures peu consommatrices...
  - ...mais très complexes à maîtriser

Des pistes de solutions

Langages spécifiques à un domaine

- haut-niveau
- confort d'utilisation
- abstraction de la cible matérielle
- exemples : SQL (bases de données), Faust (traitement du signal)

Compilation

- optimisations de code spécifiques
- différentes cibles matérielles

Sujet

La compilation face aux défis énergétiques du XXI^{e} siècle

Traitement du signal

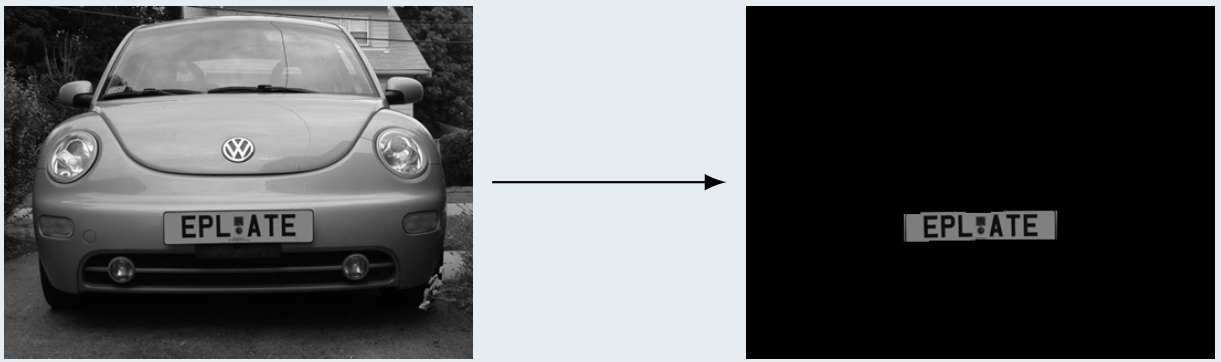
- opérateurs réguliers
- parallélisme implicite
- besoin industriel

Directeur de thèse François Irigoin
Maître de thèse Fabien Coelho
Centre de recherche en informatique, MINES ParisTech
Localisation Fontainebleau, Seine et Marne
École doctorale Sciences et Métiers de l'Ingénieur

Réalisations

Compilation vers une architecture basse consommation

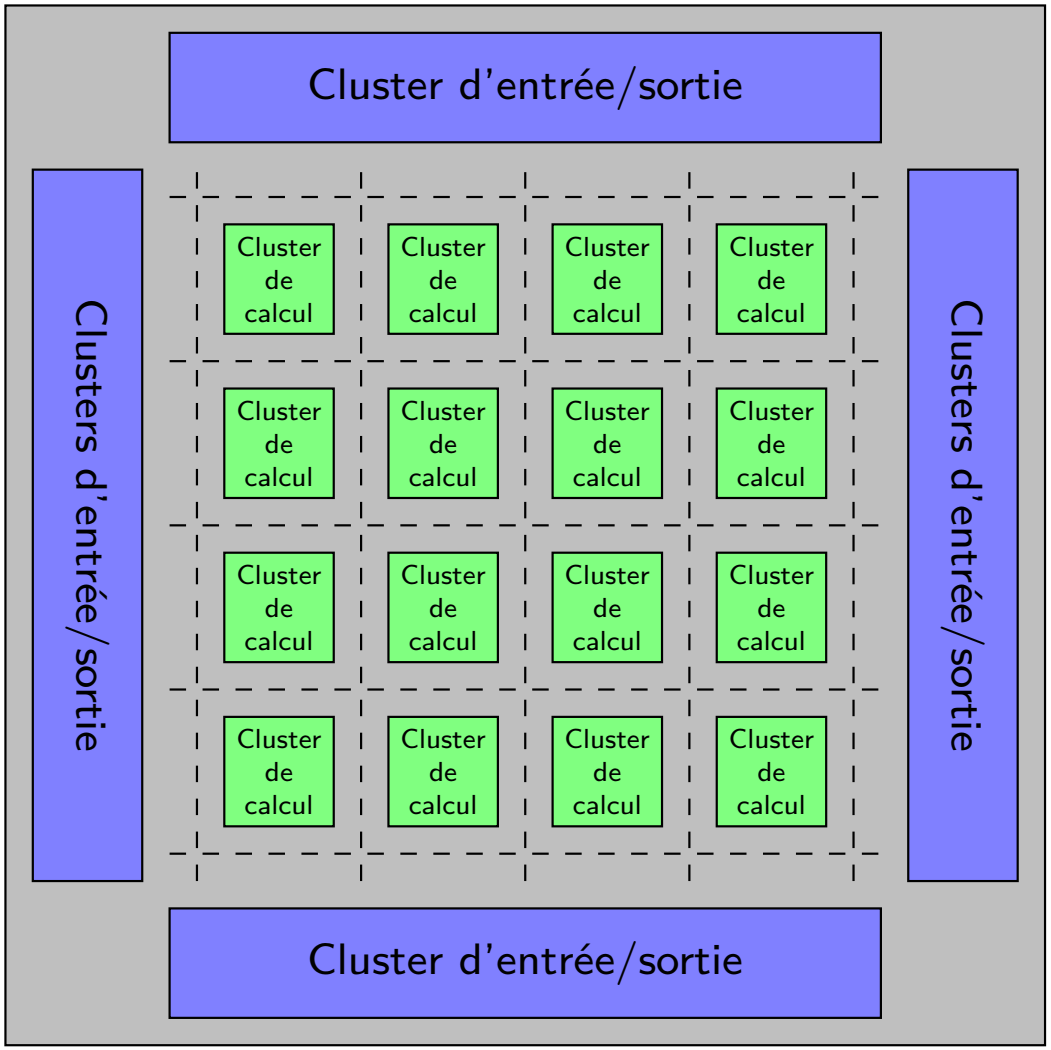
- Kalray MPPA-256
- applications d'analyse d'image



Deux étapes

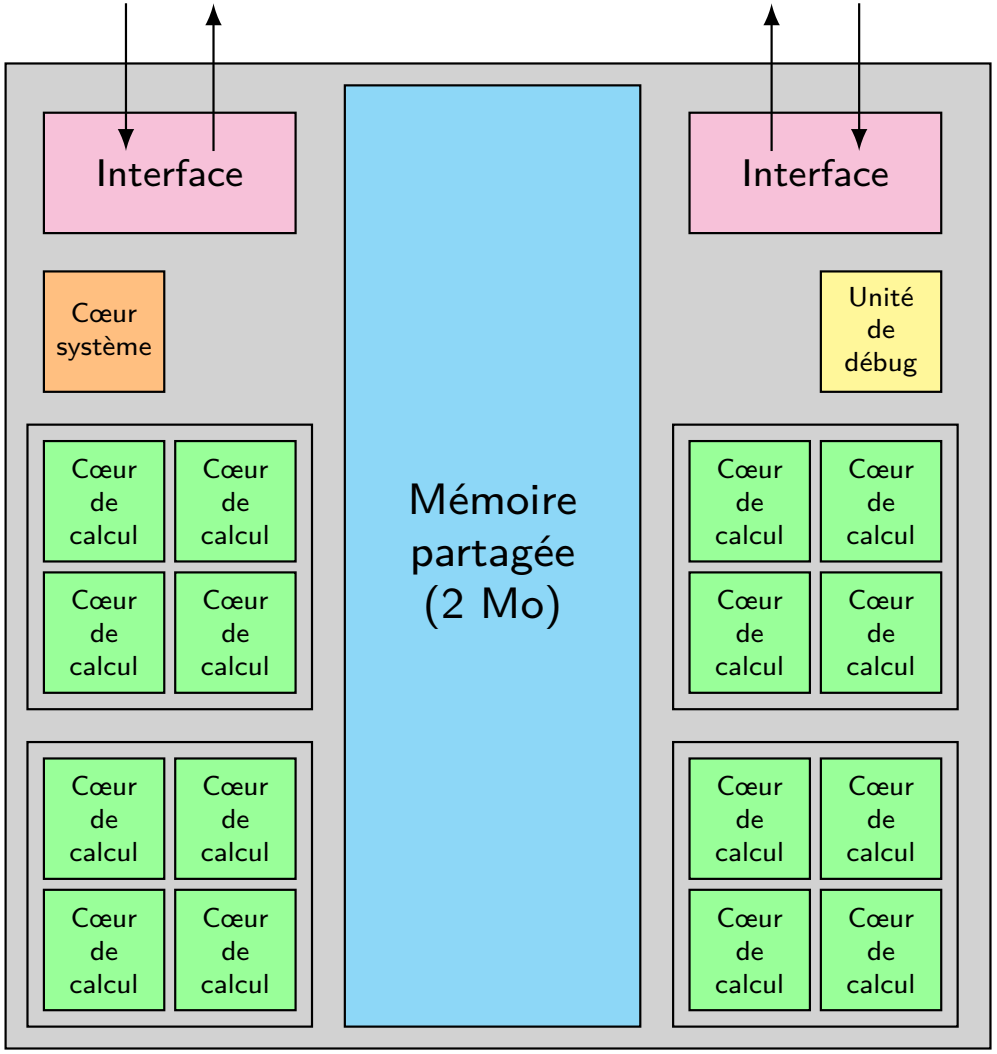
1. bibliothèque d'opérateurs de base d'analyse d'images
2. générateur de code pour le MPPA-256
   - compilateur source-à-source PIPS
   - API FREIA

Puce MPPA-256



- 16 clusters de calcul
- 4 clusters d'entrée/sortie
- 28 nm

Clusters de calcul



- 64 registres 32 bits
- VLIW + vectoriel
- Mémoire partagée : 2 Mo/cluster

Modèles de programmation

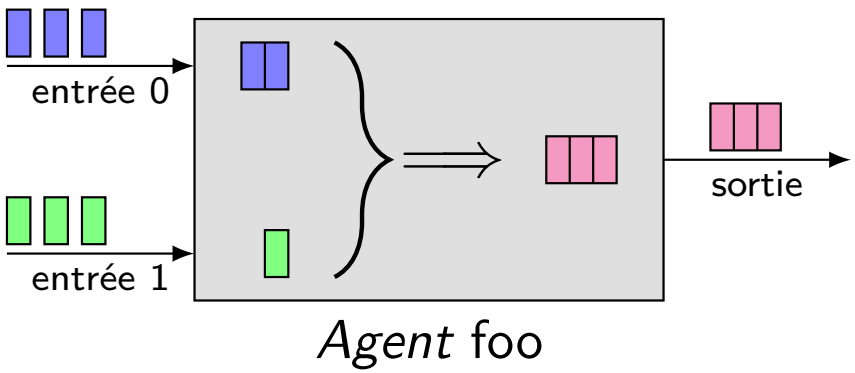
Explicitement parallèle (bas niveau)

- POSIX-Threads
- OpenMP
- MPI (portage en cours)
- OpenCL (portage en cours)

Dataflow (haut niveau)

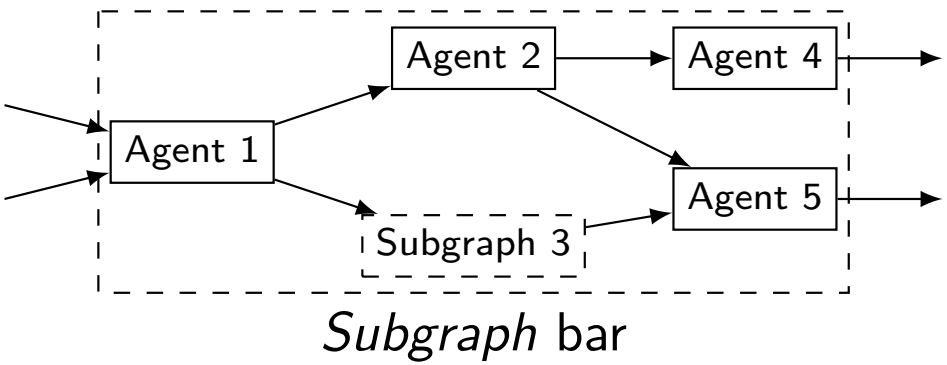
- Langage *Sigma-C*

Sigma-C : agents




```
agent foo() {
  // describe agent interface
  interface {
    in<int> input0;
    in<int> input1;
    out<int> output;
    // declare the state machine
    spec{input0[2], input1, output[3]};
  }
  // loop over the state
  void start()
    exchange (input0 inp0[2],
              input1 inp1,
              output outp[3]) {
    outp[0] = inp0[0];
    outp[1] = inp1;
    outp[2] = inp0[1];
  }
}
```

Sigma-C : subgraphs



Subgraph bar

```
subgraph bar() {
  // describe subgraph interface
  interface { ... }
  map {
    // instanciate agents
    agent a1 = new Agent1();
    ...
    agent a3 = new Subgraph3();
    // connect agents
    //to subgraph interfaces
    connect (input0, a1.input0);
    ...
    connect (a5.output, output1);
    // connect agents
    connect (a1.output0, a2.input);
    ...
    connect (a3.output, a5.input1);
  }
}
```

Bibliothèque d'agents *Sigma-C* pour l'analyse d'images

Plusieurs types d'agents développés

- opérateurs arithmétiques
  - unaires (une image en entrée + un paramètre)
  - binaires (deux images en entrée)
- opérateurs morphologiques
  - érosion, dilation, convolution
  - cœur de boucle en assembleur (développé par Kalray)
- opérateurs de mesure
  - min et max (avec ou sans leurs coordonnées), volume

$\Longrightarrow$ 3500 lignes de code *Sigma-C*

Génération de subgraphs *Sigma-C*

Principe

Entrée : code FREIA

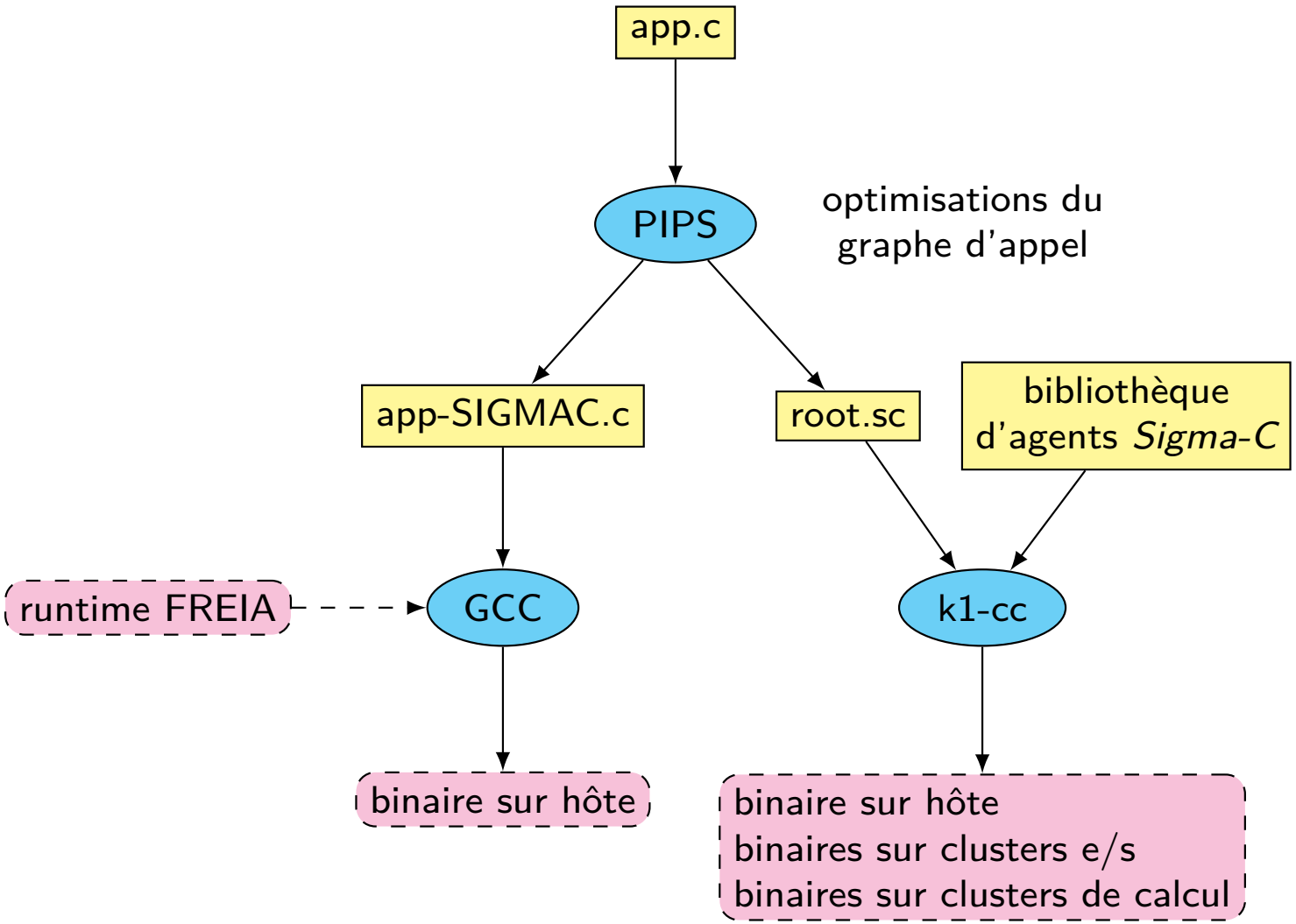
```
freia_aipo_erode_8c(imerode, imin, kernel);
freia_aipo_dilate_8c(imdilate, imerode, kernel);
freia_aipo_and(imout, imdilate, imin);
```

Sortie : subgraph *Sigma-C*

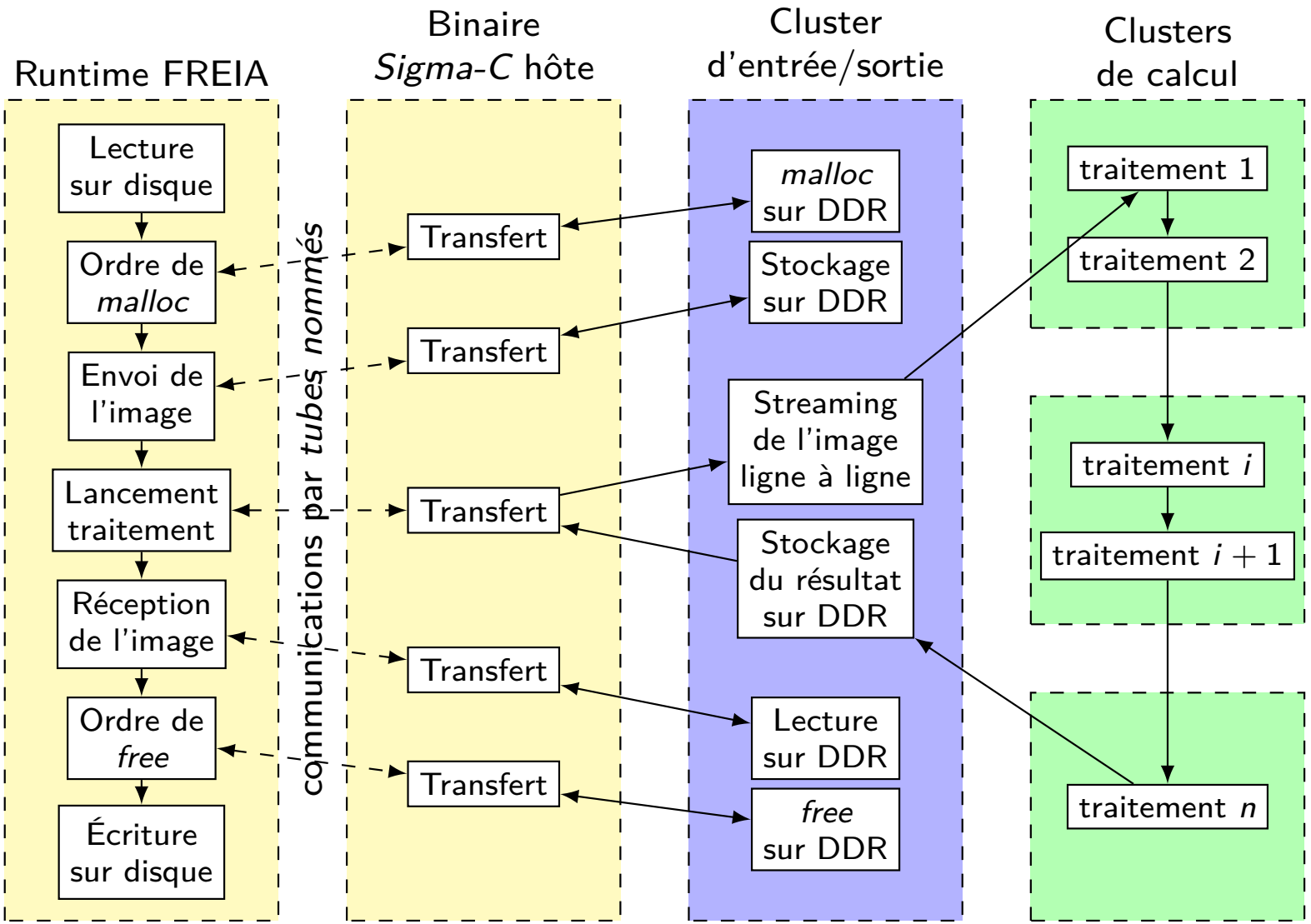
```
subgraph foo() {
  int16_t kernel[9] = {0,1,0, 0,1,0, 0,1,0};
  ...
  agent ero = new img_erode(kernel);
  agent dil = new img_dilate(kernel);
  agent and = new img_and_img();
  ...
  connect(ero.output, dil.input);
  connect(dil.output, and.input);
  ...
}
```

$\Longrightarrow$ 1100 lignes de code C

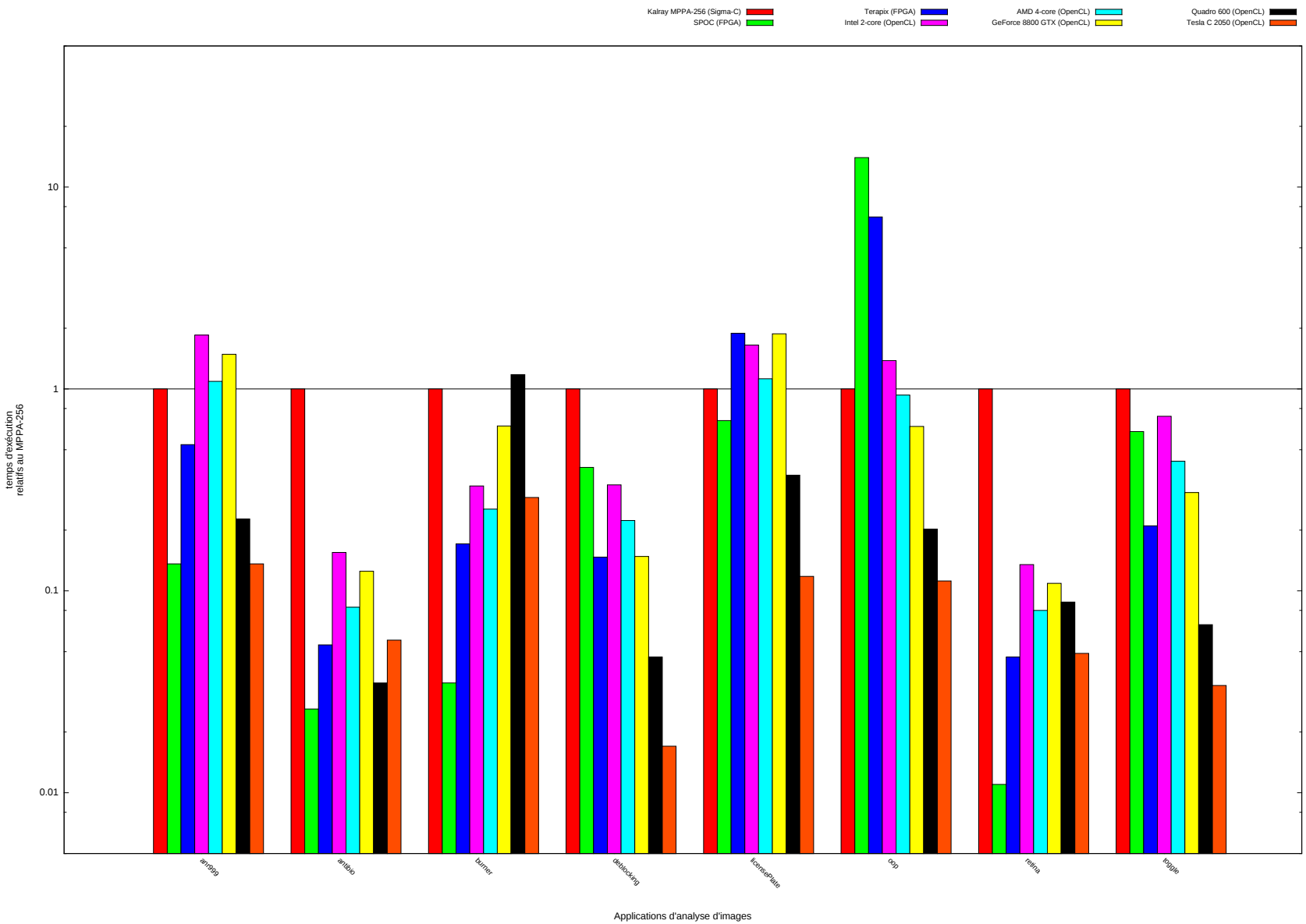
Phases de compilation



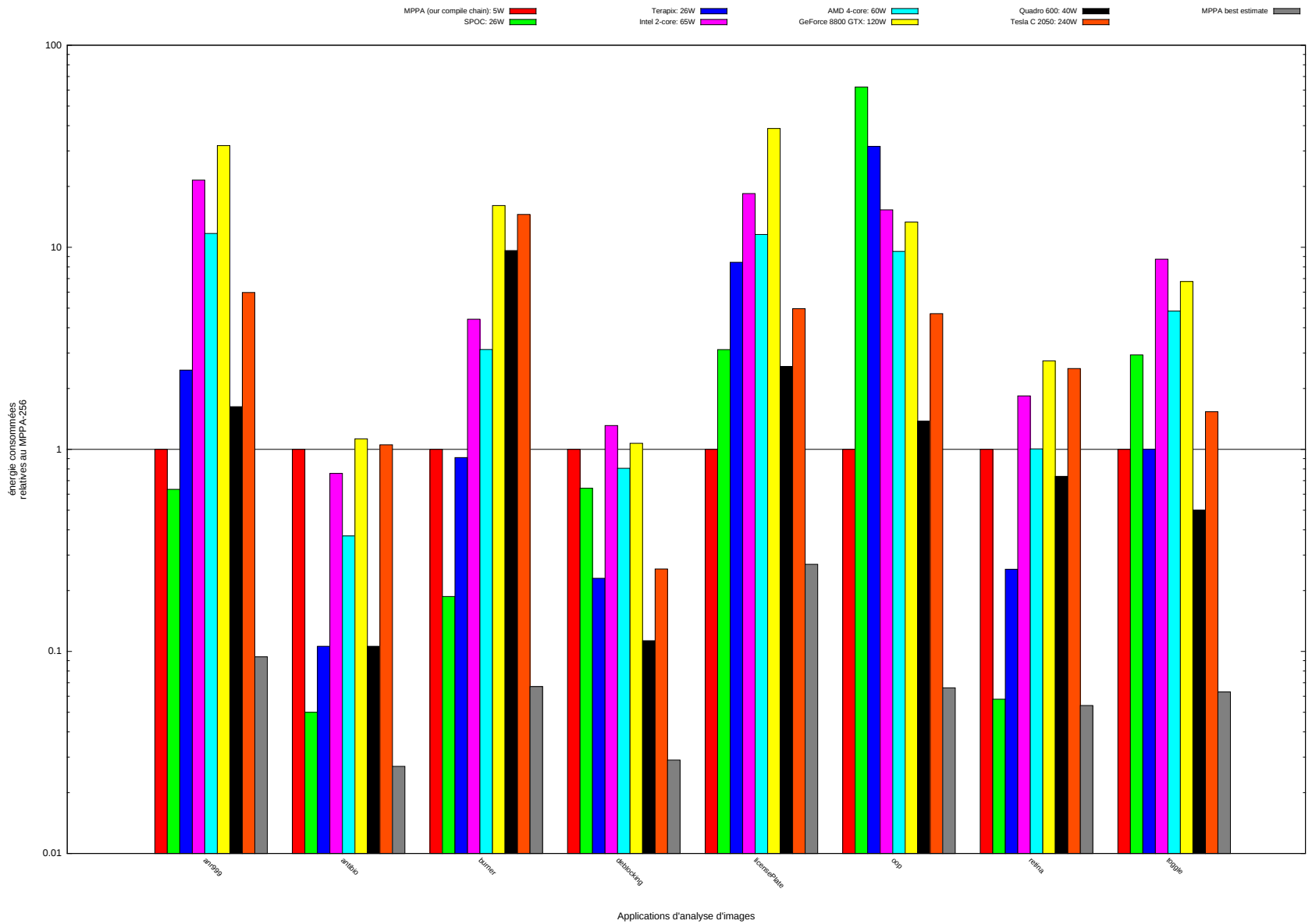
Environnement d'exécution



Comparaison avec d'autres plateformes matérielles



Comparaison avec d'autres plateformes matérielles



Bibliographie

Techniques de compilation

- Compilers : Principles, Techniques and Tools, *Aho, Lam, Sethi et Ullman*
- Supercompilers for Parallel and Vector Computers, *Zima et Chapman*

Compilation vers accélérateurs matériels

- Transformations de programme automatiques et source-à-source pour accélérateurs matériels de type GPU, *Amini*
- Abstractions performantes pour cartes graphiques, *Bourgoin*
- API compilation for image hardware accelerators, *Coelho et Irigoin*

Consommation énergétique

- The Energy/Frequency Convexity Rule : Modeling and Experimental Validation on Mobile Devices, *De Vogeleer et al.*

Bibliographie

Transformations de code

- Contribution à l'optimisation de programmes scientifiques, *Zory*
- Building Source-to-Source Compilers for Heterogenous Targets, *Guelton*
- Tiling Stencil Computations to Maximize Parallelism, *Bandishti et al.*

Gestion des mémoires

- Optimisation des transferts de données pour le traitement du signal : pavage, fusion et réallocation des tableaux, *Bouchebaba*

Traitement d'images

- Halide : A Language and Compiler for Optimizing Parallelism, Locality, and Recomputation in Image Processing Pipelines, *Ragan-Kelley et al.*

Conférences et séminaires

Participations

7es Journées de la Compilation décembre 2013 **présentation**
Seine et Marne

Séminaire mensuel du CRI avril 2014 **présentation**

ComPAS'2014 avril 2014, Neuchâtel, Suisse **poster**

Séminaire Parkas Alain Darte, modèle polyédrique, février 2014

Séminaire langages streaming et dataflow avril 2014,
St Germain du Mont d'Or, Rhône

Soutenances de thèse et HDR

- Fabien Coelho (CRI, MINES ParisTech)
- Dounia Khaldi (CRI, MINES ParisTech)
- Mathias Bourgoin (LIP-6, UPMC)
- Feng Li (LIP-6, UPMC)

Formation doctorale

Modules de formation professionalisants

Intelligence Économique et Stratégique	session mars 2014	21h
Lecture rapide	session mars 2014	21h

Total : 42 heures sur 60 demandées

Anglais

TOEIC (mai 2013) 950/990

Activités d'enseignement

Cycle ingénieurs civils

Travaux pratiques

ES Applications Réparties	3e année – réseaux	3h30
ES Systèmes d'Information	2e année – bases de données	6h

Conclusion

Traitement d'images sur architecture manycore

- plusieurs applications fonctionnelles
- résultats à optimiser
- présentations à des conférences

Court terme : objectif publication

- déroulement de boucles ? — phase de préparation
- fusion d'opérateurs arithmétiques ? — DAG
- occupation de tous les cœurs de calcul ? — runtime

Moyen terme : extension vers le traitement du signal

- opérateurs de traitement du signal
- trouver les transformations pertinentes
- définir un DSL (VSIPL/Thales)

Compilation d'applications de traitement du signal sur accélérateurs matériels à haute efficacité énergétique

Rapport d'avancement 1re année

Pierre Guillou

Directeur de thèse : François Irigoin
Maître de thèse : Fabien Coelho
MINES ParisTech – Centre de recherche en informatique

26 mai 2014