DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

Modélisation de la performance et optimisation d'un algorithme hydrodynamique de type Lagrange-Projection sur processeurs multi-cœurs

T. Gasc[1,2,3], F. De Vuyst[1], R. Motte[3], M. Peybernes[4], R. Poncet[5]

[1] CMLA UMR 8536 ENS CACHAN, France
[2] Maison de la Simulation USR 3441, CEA Saclay, France
[3] CEA, DAM, DIF, F-91297 Arpajon, France
[4] CEA Saclay, DEN, F-91191 Gif-sur-Yvette, France
[5] CGG, 27 Avenue Carnot, 91300 Massy, France

MAISON DE LA SIMULATION

CMLA
Centre de Mathématiques et de Leurs Applications

Séminaire Aristote
5 février 2015

DE LA RECHERCHE À L'INDUSTRIE
cea

Plan

Objectifs des modèles des performances

Étudier du point de vue HPC une méthode numérique

Analyser et comprendre la performance de méthodes numériques

... adapter une méthode numérique ...

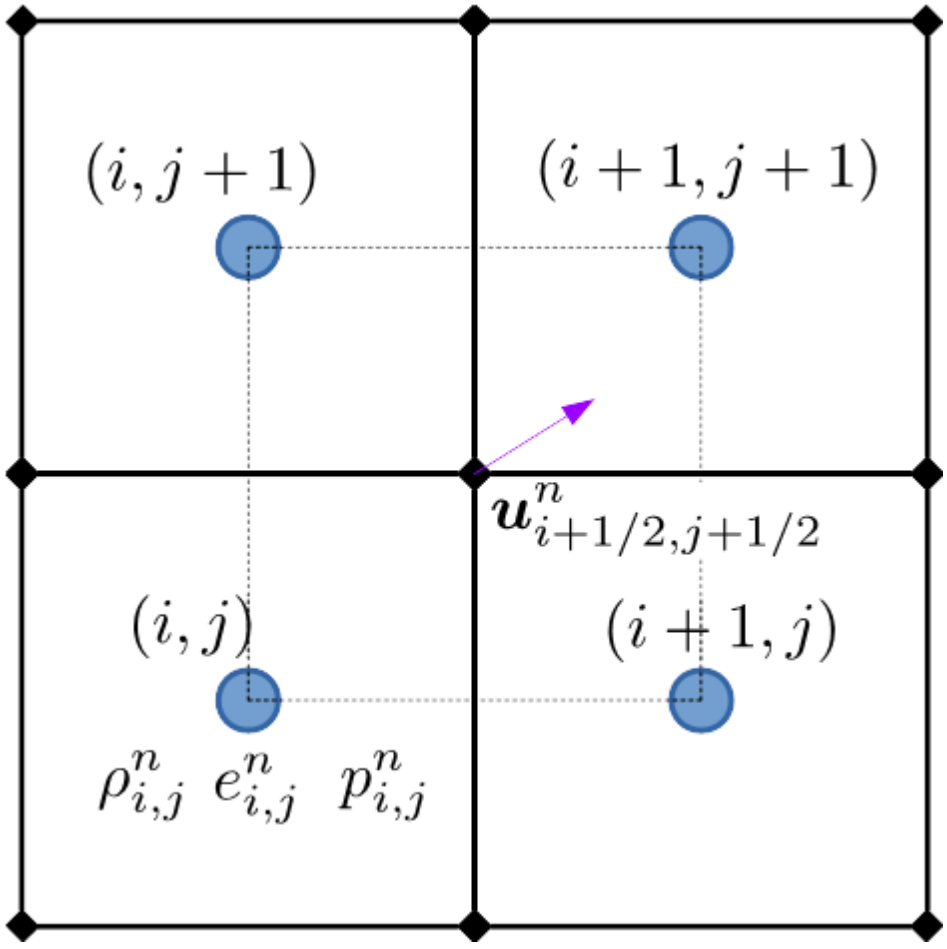
Comprendre les architectures actuelles
Optimiser les algorithmes pour ces architectures

... vers l'exaflop

Prévoir et anticiper les architectures futures
Proposer d'autre(s) alternative(s) (méthodes numériques) plus performante(s)

Méthode numérique d'étude : solveur hydrodynamique 1/3

(version à variables décalées, variables de vitesse)



[Hirt et Amsden, 1974] [Collela et Woodward, schéma BBC, 1984]

Méthode numérique d'étude : solveur hydrodynamique 2/3

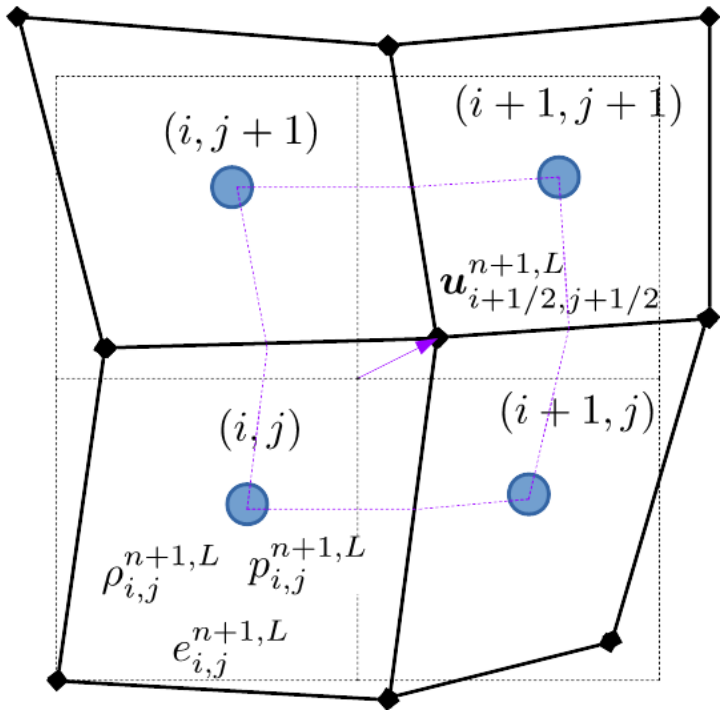
1 - Phase lagrangienne

Évolution des grandeurs d'intérêt sur un pas de temps sur le maillage lagrangien

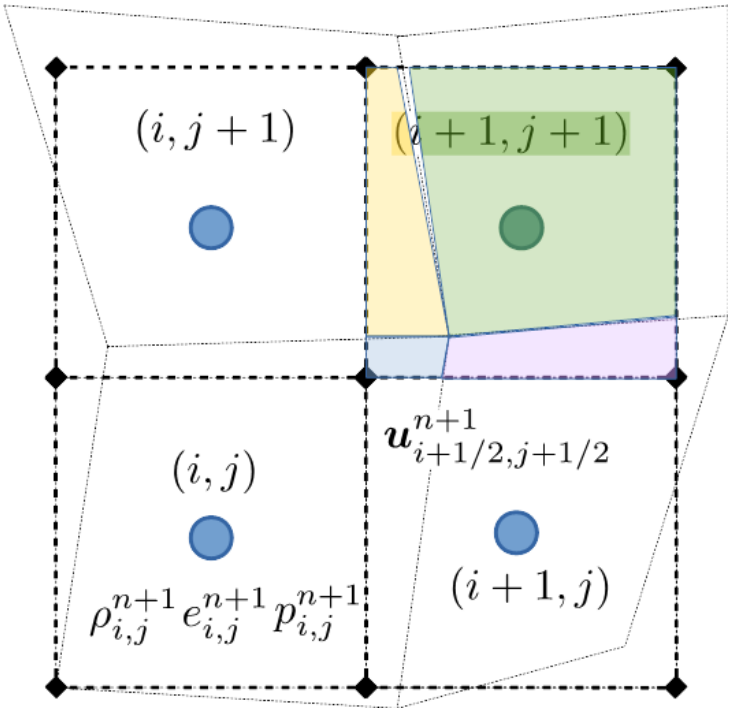
$$\rho D_t \tau = \nabla \cdot \mathbf{u},$$

$$\rho D_t \mathbf{u} + \nabla p = 0,$$

$$\rho \left(D_t e + p D_t \tau \right) = 0.$$



Méthode numérique d'étude : solveur hydrodynamique 3/3



2 - Projection

Projection / interpolation du maillage déformé sur le maillage initial

$$\partial_t \rho + \mathbf{u} \cdot \nabla \rho = 0,$$

$$\partial_t (\rho \mathbf{u}) + \mathbf{u} \cdot \nabla (\rho \mathbf{u}) = 0,$$

$$\partial_t (\rho e) + \mathbf{u} \cdot \nabla (\rho e) = 0.$$

Graphe des kernels du solveur hydrodynamique (1/2)

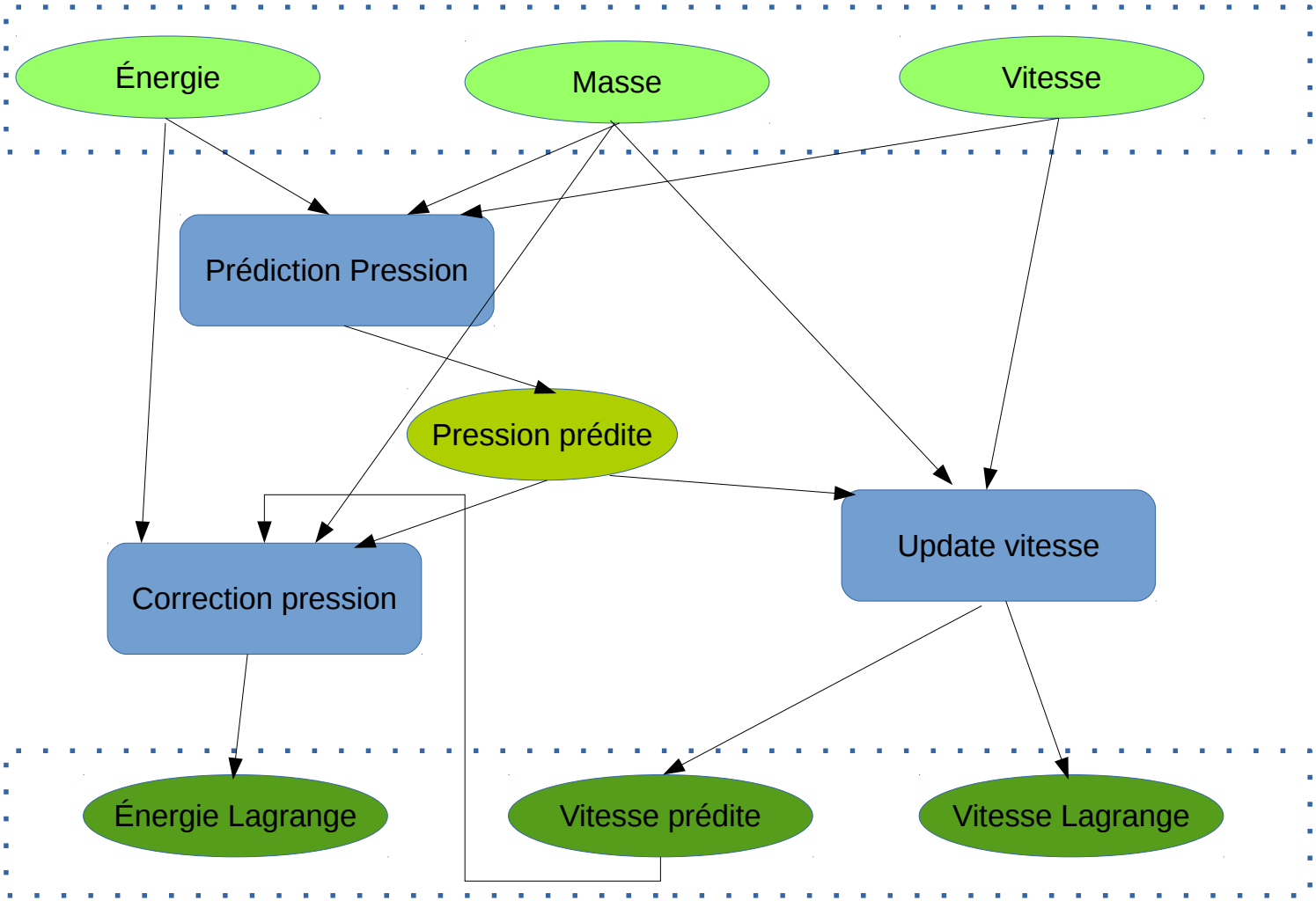


Figure : Graphe de dépendance des kernels du solveur hydrodynamique : phase 1 - Lagrange

Graphe des kernels du solveur hydrodynamique (2/2)

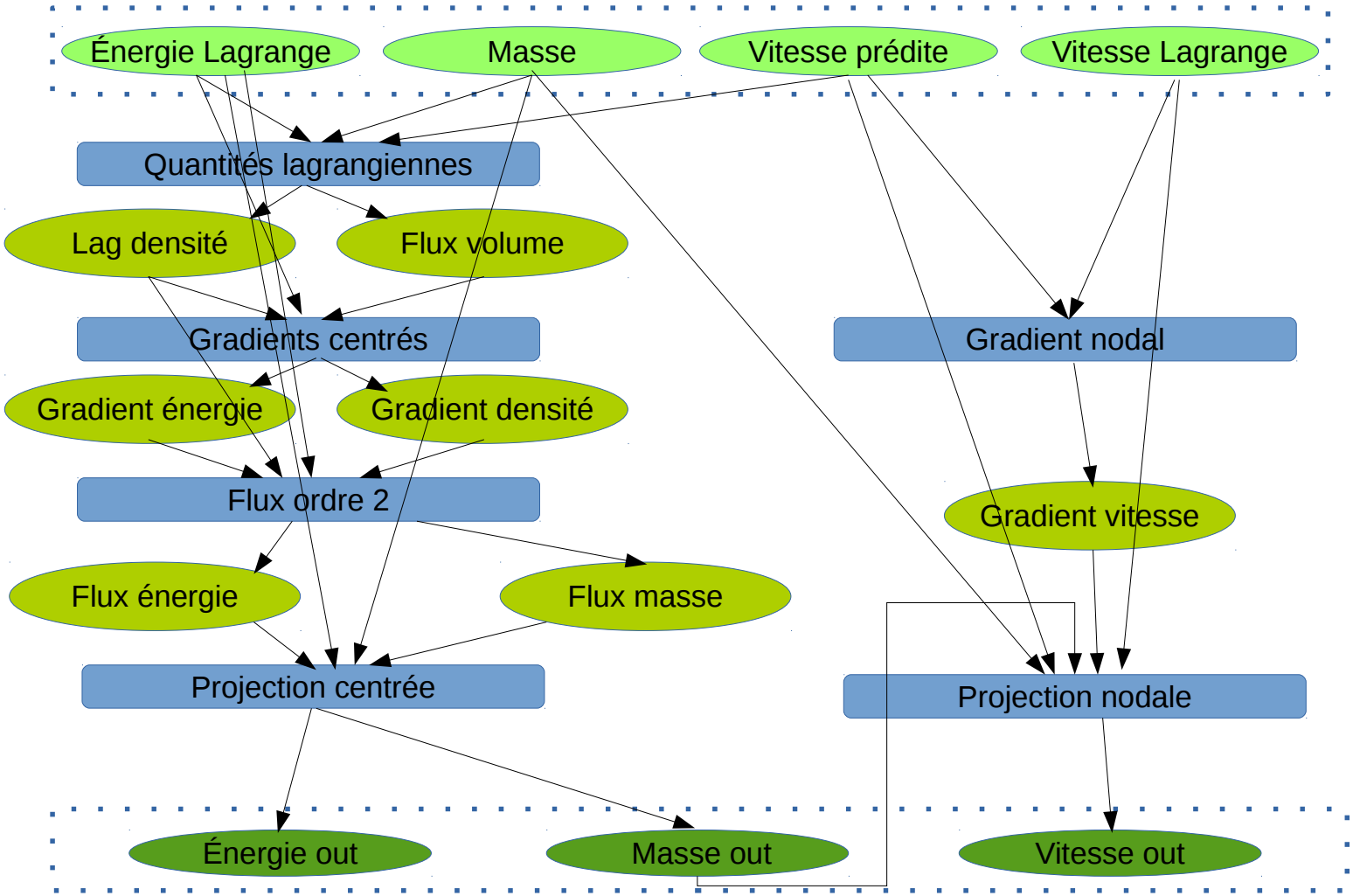


Figure : Graphe de dépendance des kernels du solveur hydrodynamique : phase 2 - Projection

Plan

Modélisation de la performance

Objectif / définition

Construire et utiliser une abstraction simple qui permet de prédire approximativement la vitesse d'exécution d'une méthode numérique donnée sur une architecture donnée.
$\Rightarrow$ Être en mesure d'estimer et de comprendre la performance atteignable d'une méthode numérique sur une machine donnée.

Intérêts

- Situer l'efficacité de l'implémentation par rapport à la machine
- Identifier des voies d'optimisation
- Comparer les performances de plusieurs méthodes numériques
- Prédire / anticiper la performance d'un code sur une architecture future

Roofline model : métriques

Architecture

- **Bande passante (bandwidth)** :
  vitesse de transfert des données depuis la mémoire vers les unités de calcul [Bytes/s]
- **Peak** :
  limite de performance théorique de la machine [Flops]

Algorithme

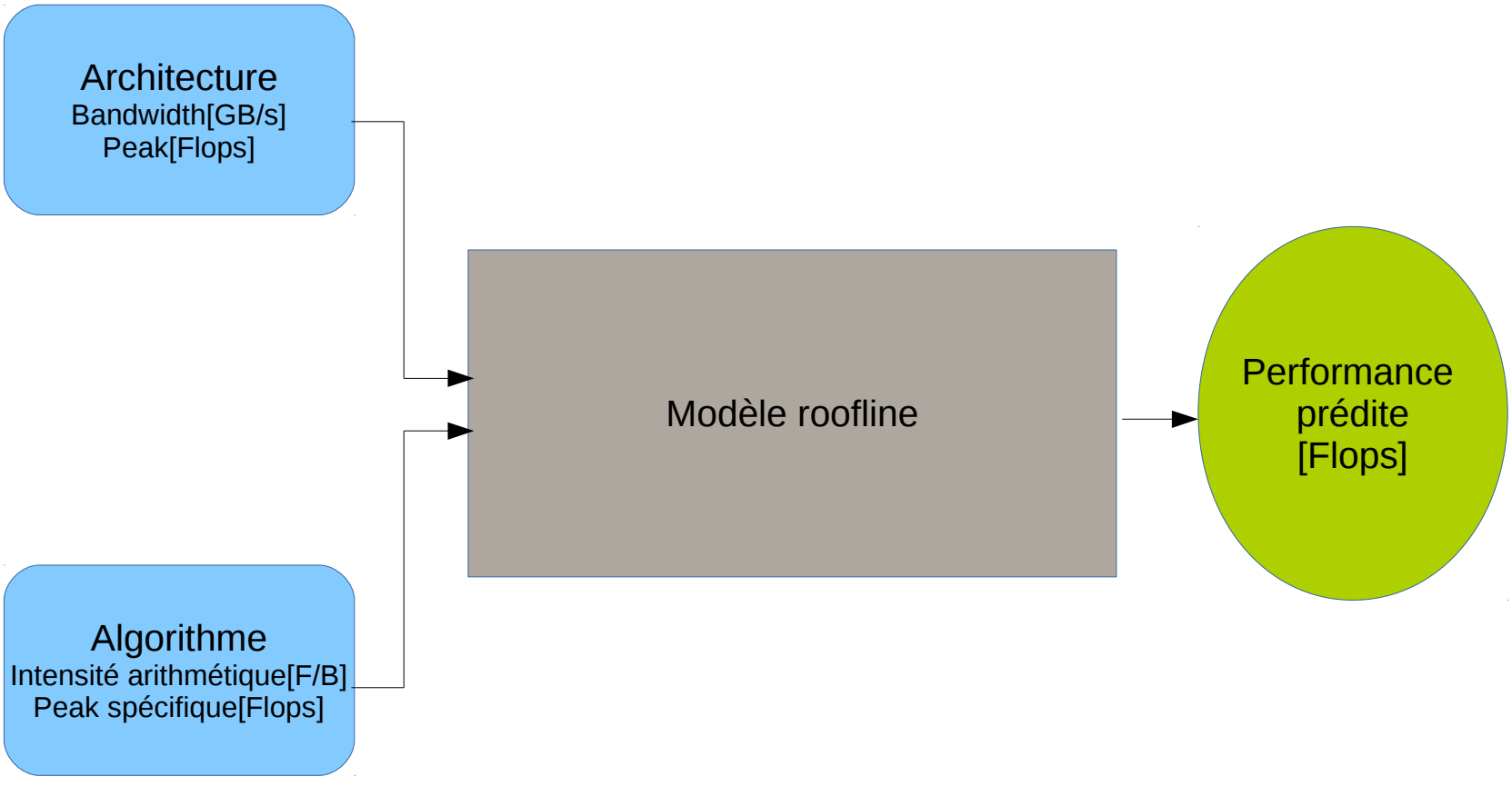
- **Intensité arithmétique (IA)** :
  $IA := \frac{\text{nombre d'opérations}}{\text{quantité de données transférées}}$ [Flops/Byte]
- **Peak spécifique** :
  vitesse maximale d'exécution de l'algorithme [Flops]

Roofline : définition

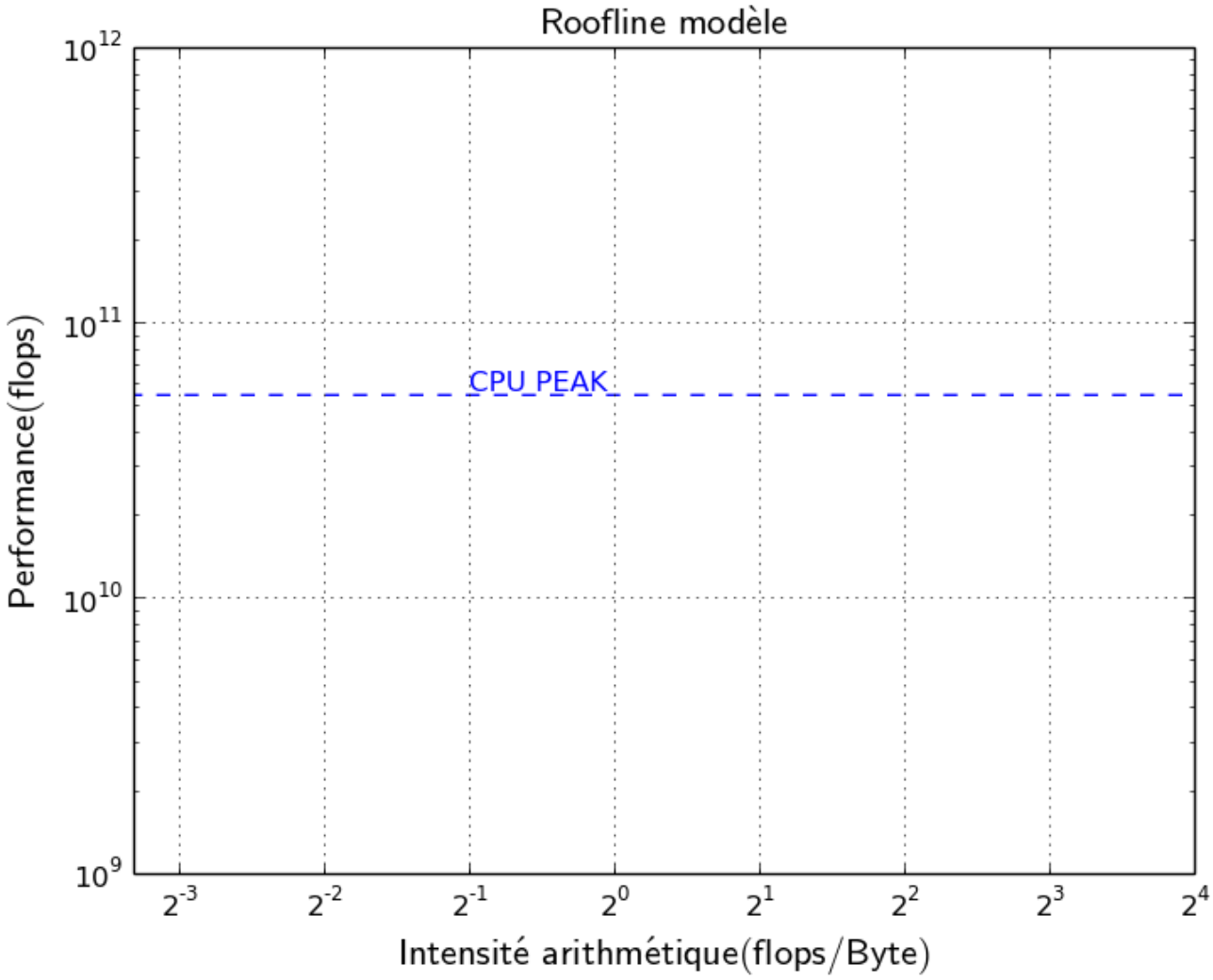
Roofline model

$$\begin{aligned} \text{Performance} &= f(Peak, bandwidth, IA) \\ &:= min(Peak, bandwidth \times IA) \end{aligned}$$

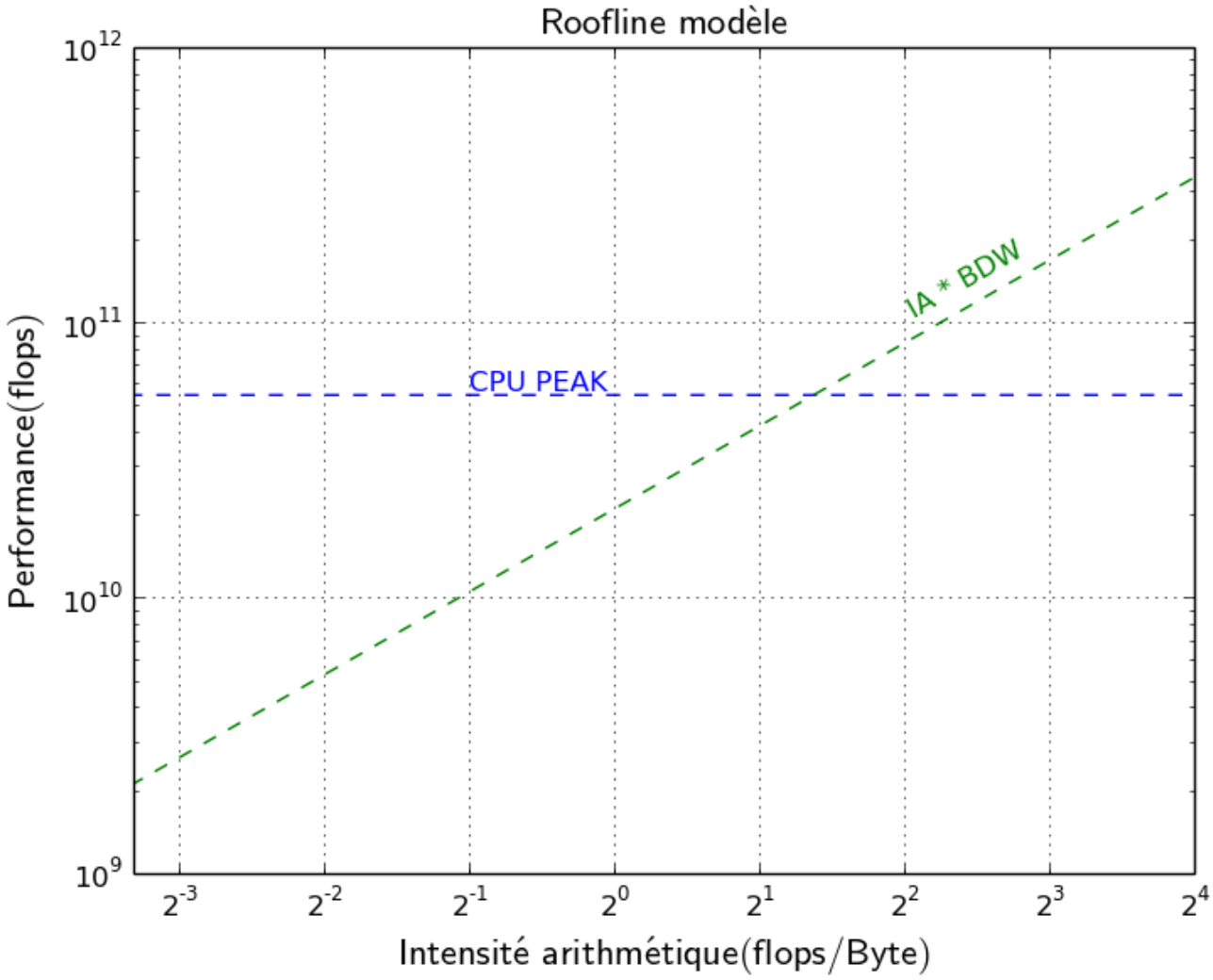
Roofline : principe



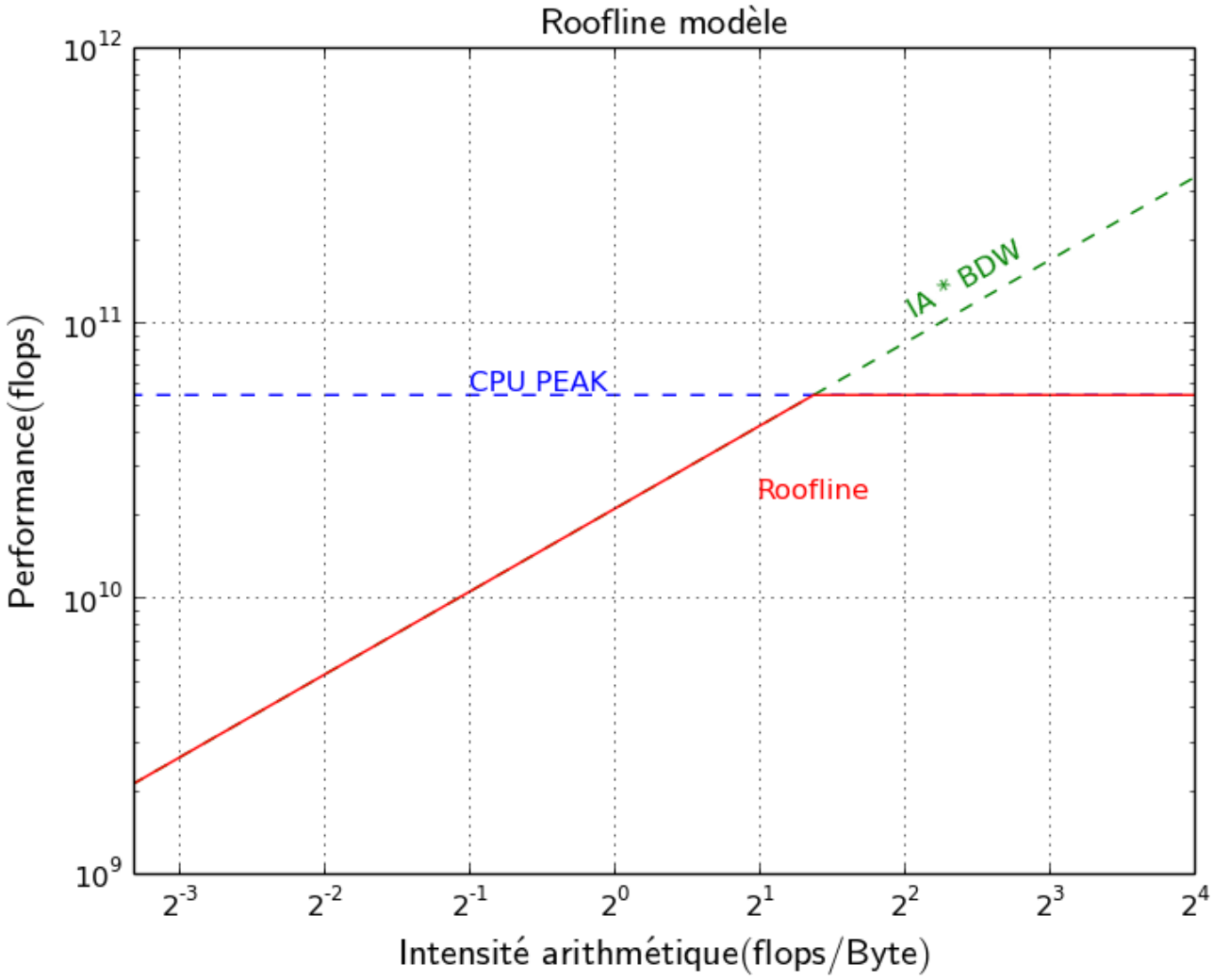
Roofline : principe graphique (1/3)



Roofline : principe graphique (2/3)



Roofline : principe graphique (3/3)



Éléments bibliographiques sur les modèles de performance

Modélisation

Roofline : [Williams et al, 2009]
Execution Cache Memory model [ECM] : raffinement du roofline pour les IA faibles [Treibig & Hager, 2010]

Utilisation du modèle (modélisation et optimisation)

- Opérations élémentaires : DGEMM, SpMV
- Bibliothèques : 3dFFT, FMM
- Applications légères (mono kernel) : Stencil, Jacobi, LBM, imagerie médicale [Hager,Treibig et al., 2006, 2013]
- Travaux sur co-processeurs :
  fluide incompressible sur GPU [Etancelin, 2014],
  advection sur Intel MIC [A. Mrabet, P. Thierry, J.M. Ghidaglia]

Plan

Protocole expérimental

Outils

`likwid`

`IACA`

Architectures de test

SandyBridge i3-2130 (2 cœurs/ 4 threads)

Haswell i5-4590T (4 cœurs/ 4 threads)

Code

`Shy` `https://github.com/RaphaelPoncet/shy`

Roofline modèle (1/2)
Comparaison modèle vs mesure

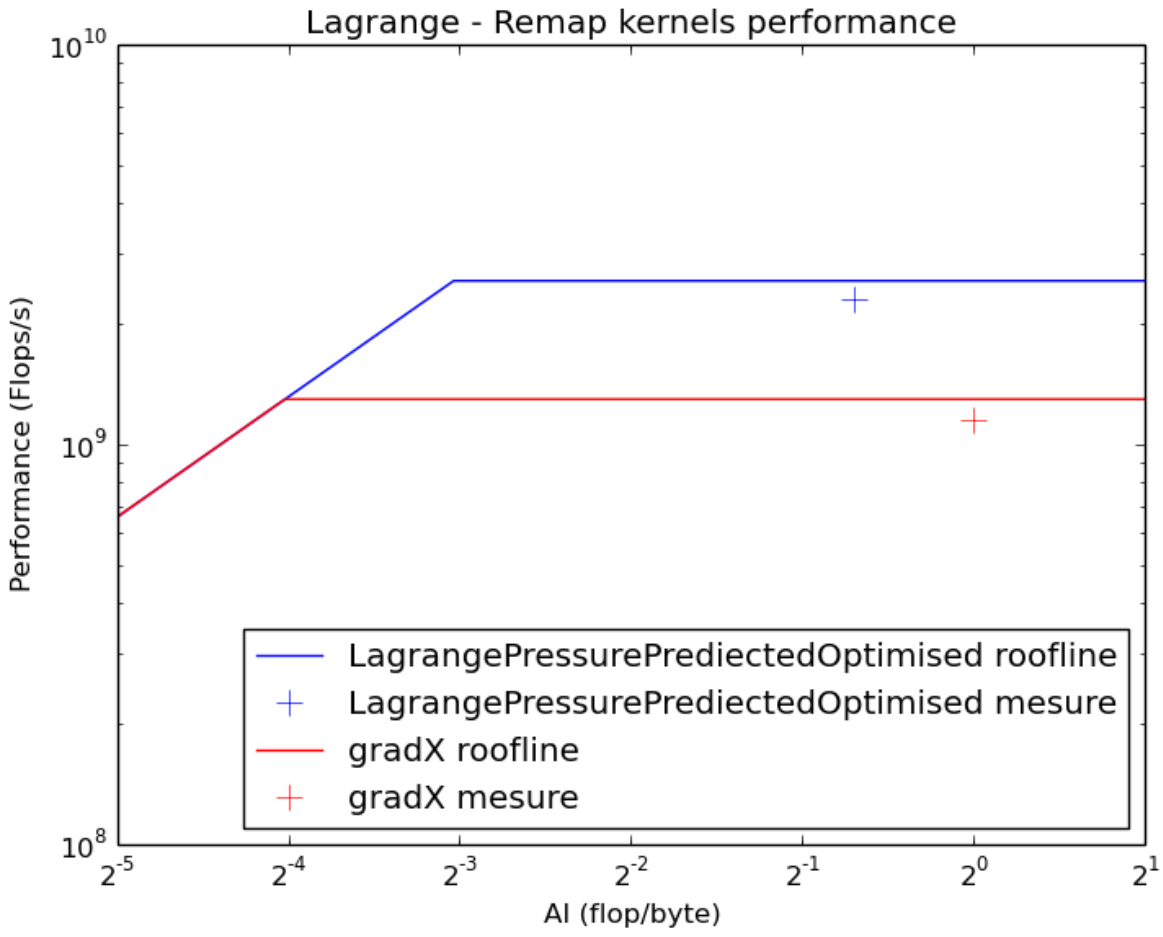


Figure : roofline pour différents kernels

Roofline modèle (2/2)

Comparaison modèle vs mesure

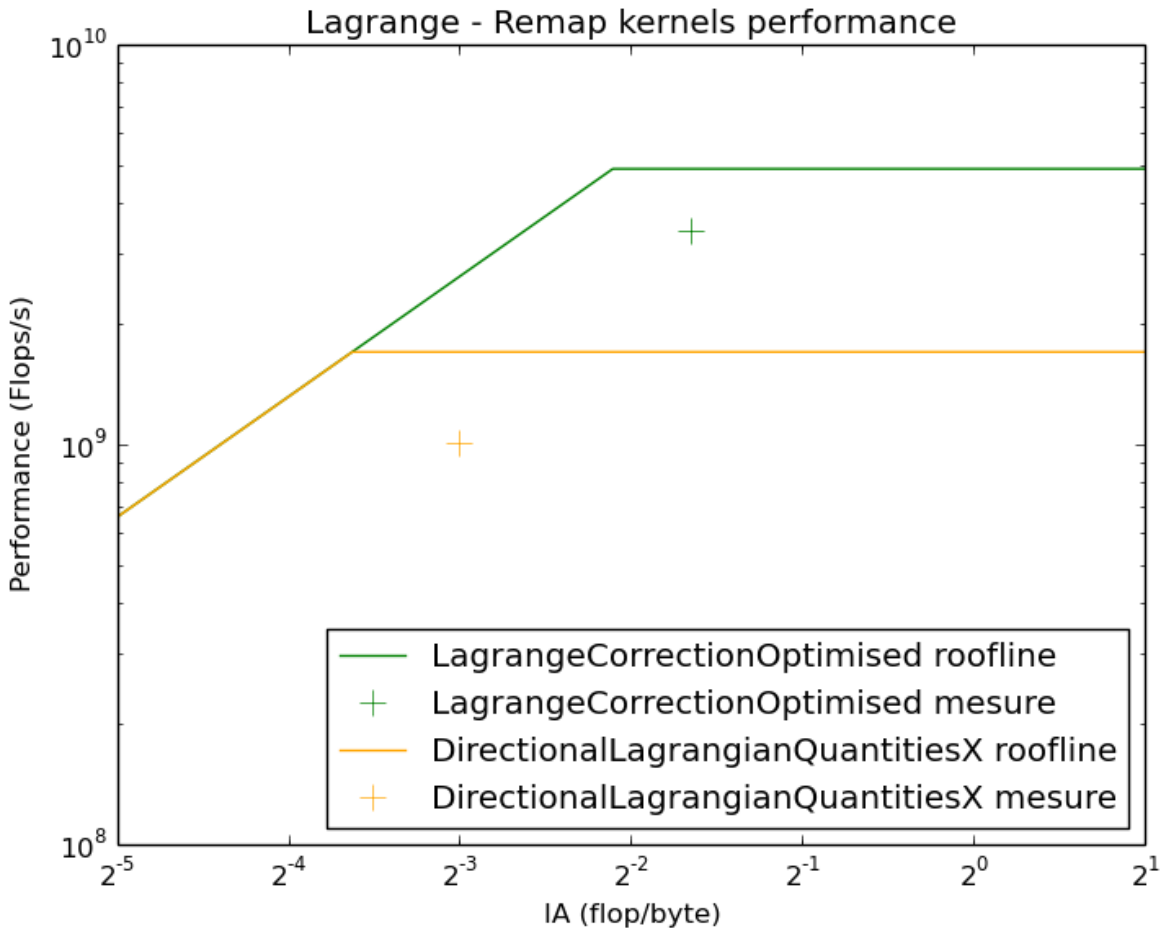
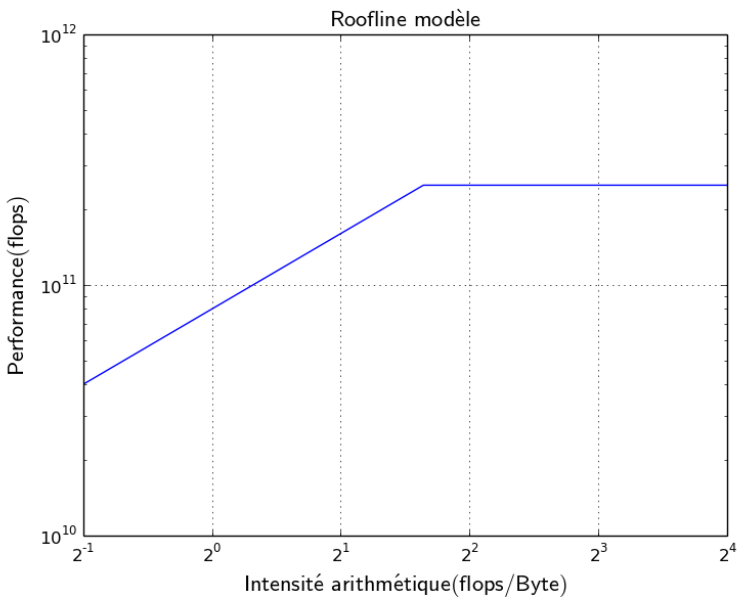
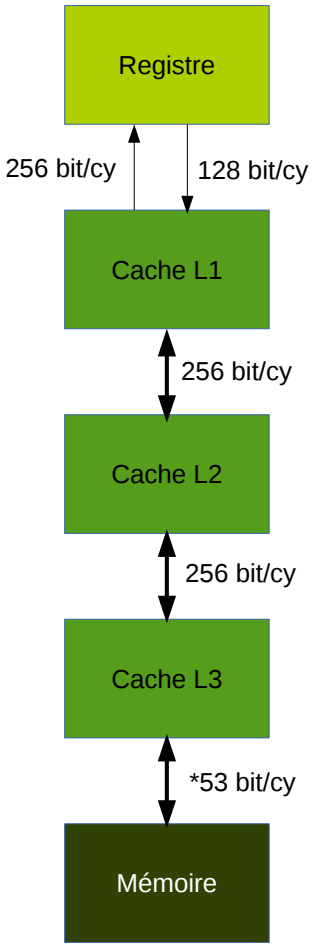


Figure : roofline pour différents kernels

Limites du Roofline

- Le comportement mémoire ne traduit pas la complexité du hardware (niveau de cache)
- Les points sous le coin du toit sont mal prédits





Nouvelle valeur de prediction : ECM (1/2)

Comparaison roofline, ECM vs mesure

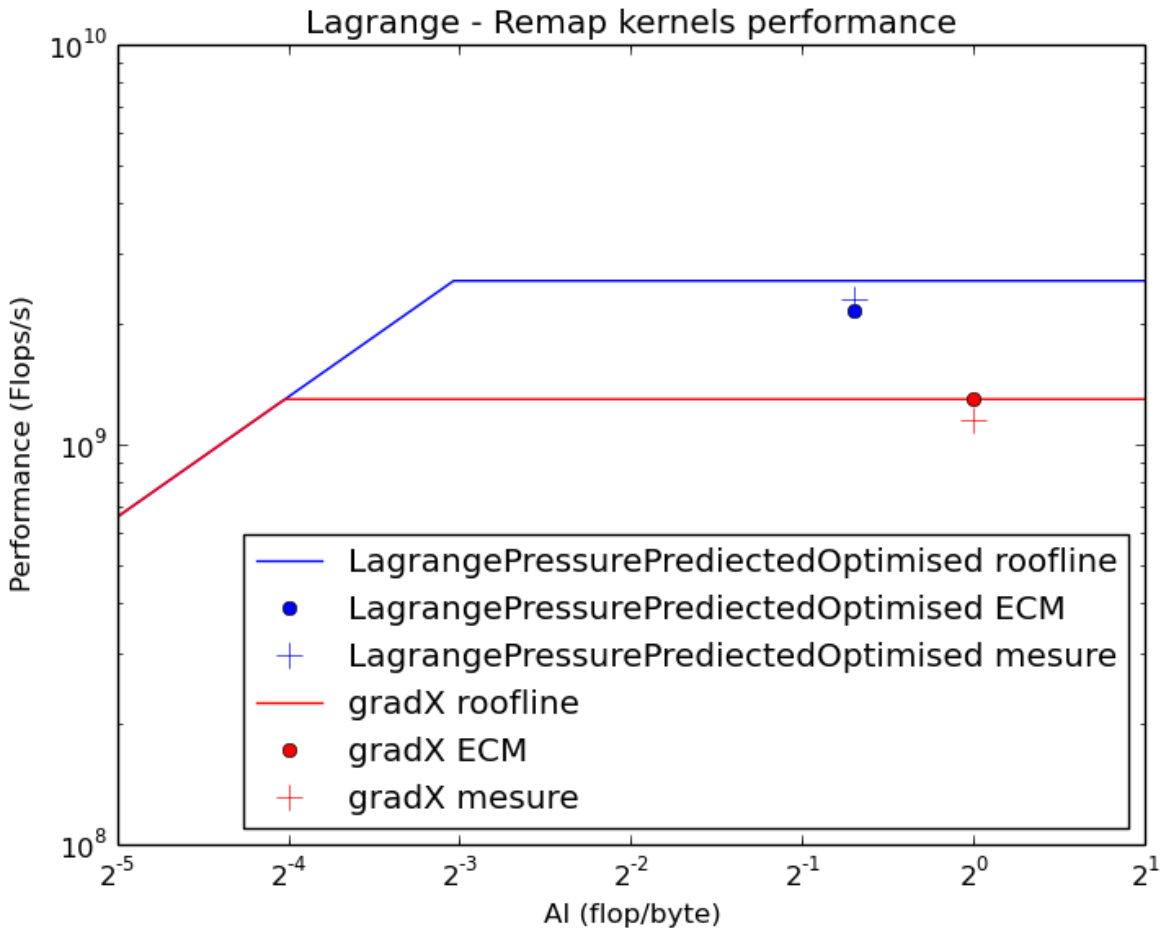


Figure : Roofline et ECM pour plusieurs kernels indépendants des caches

Nouvelle valeur de prediction : ECM (2/2)

Comparaison roofline, ECM vs mesure

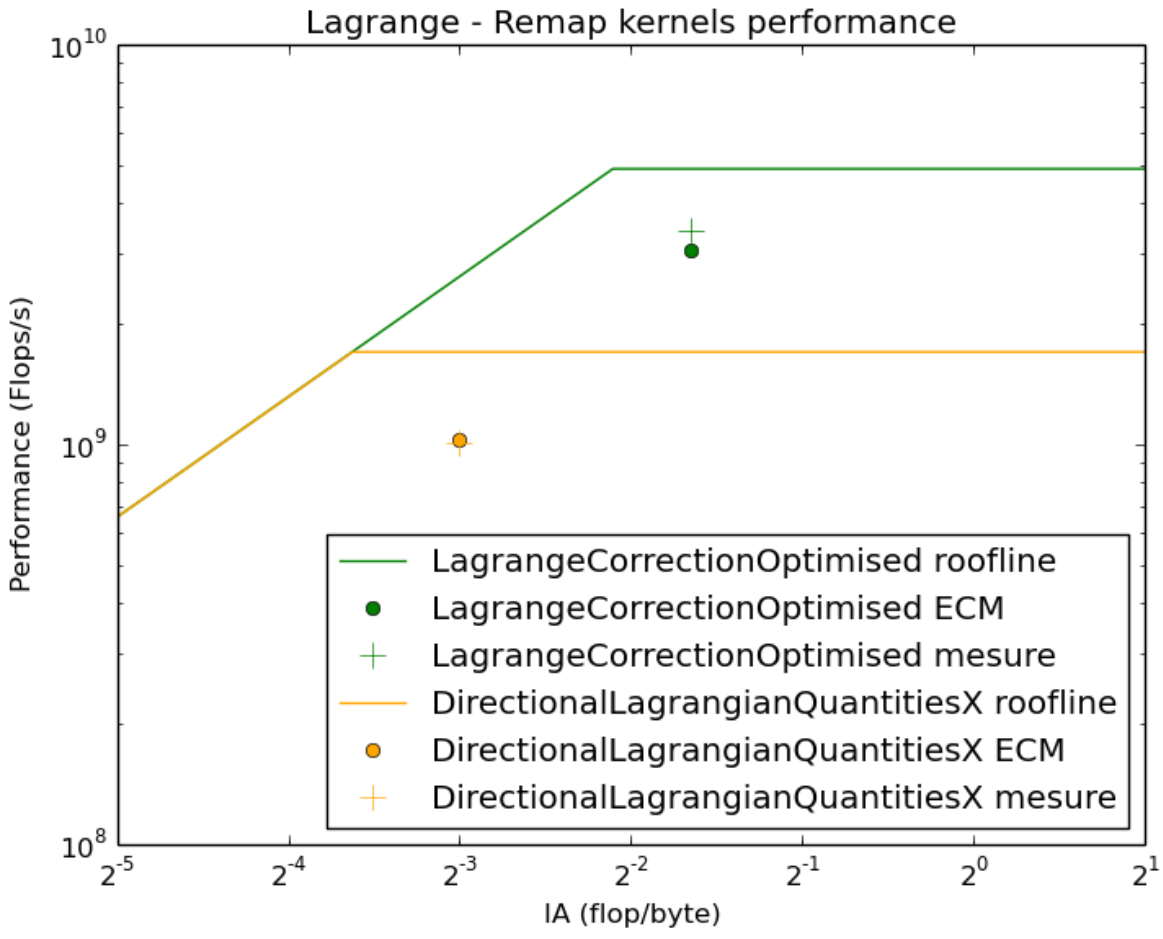


Figure : Roofline et ECM pour plusieurs kernels avec effets mémoire

ECM (1/2)
Comparaison modèle vs mesure

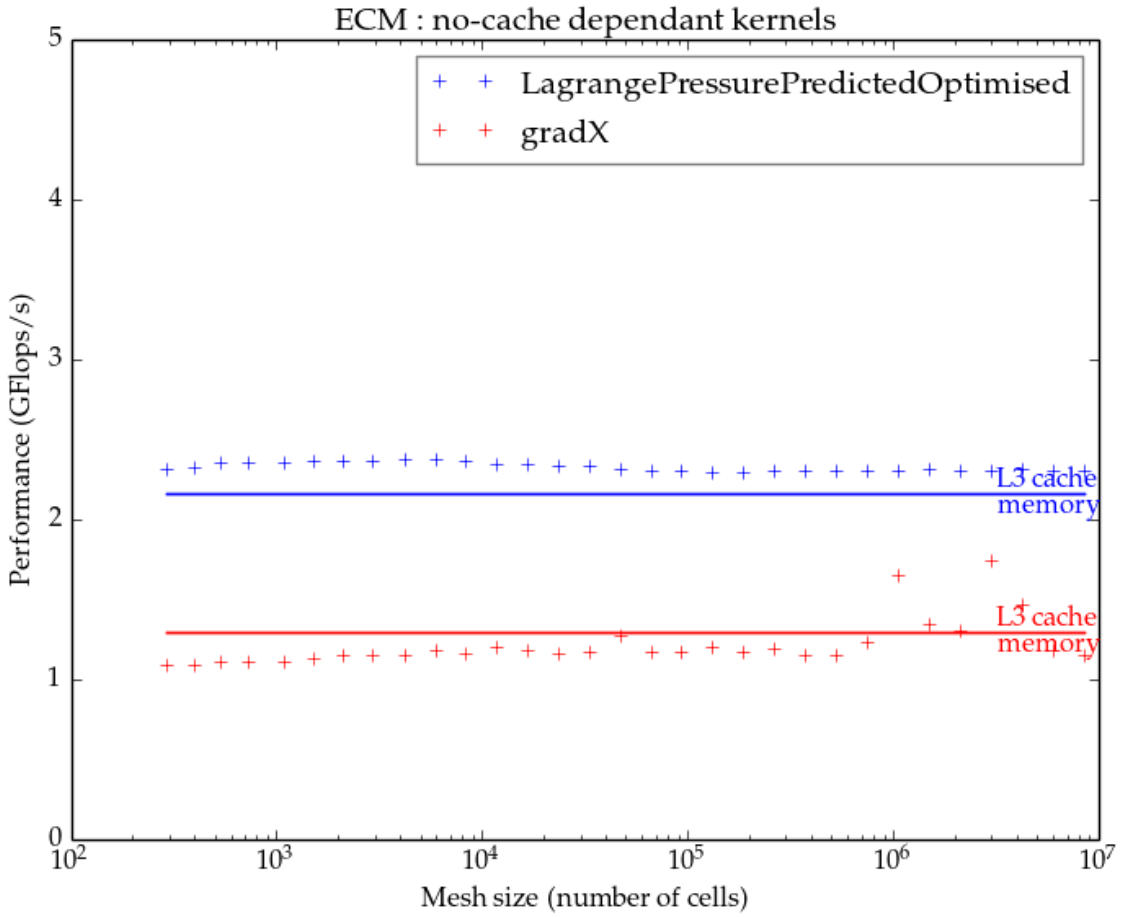


Figure : ECM pour plusieurs kernels indépendants des caches

ECM (1/2)
Comparaison modèle vs mesure

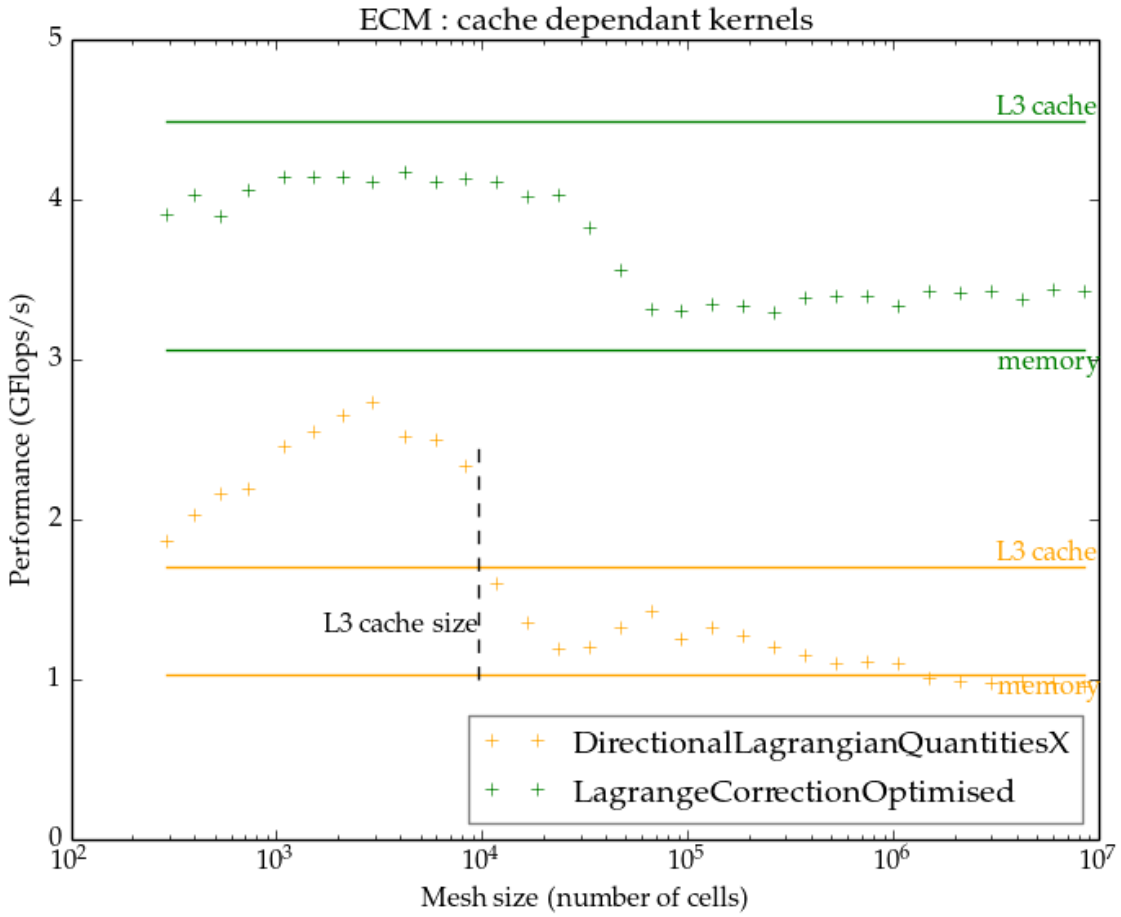


Figure : ECM pour plusieurs kernels dépendants des caches

Bilan sur l'application cible

Intérêt des modèles

- Modélisation fine du comportement du code
- Caractérisation quantitative de la vitesse d'exécution
- Possibilité de prédiction sur une architecture différente
- Identification fine des bottlenecks

Bilan sur l'application cible

Intérêt des modèles

- Modélisation fine du comportement du code
- Caractérisation quantitative de la vitesse d'exécution
- Possibilité de prédiction sur une architecture différente
- Identification fine des bottlenecks

Aller encore plus vite

- Identifier les aspects limitants la performance de la méthode numérique
- Concevoir une nouvelle méthode numérique en prenant soin d'exclure les limites identifiées
- Construire le modèle de performance pour cette nouvelle méthode
- Comparer les méthodes, idéalement sur plusieurs architectures

Conclusions

Ce qui a été présenté

- Introduction à la modélisation de performance
- Présentation de deux modèles
- Utilisation des modèles sur une application industrielle (code hydro)

Ce que vous pouvez retenir pour d'autres applications

- Les modèles simples donnent de bonnes indications qualitatives (capacité machine, distance au peak, voies potentielles d'optimisation)
- La modélisation / prédiction fine requiert une très bonne connaissance du hardware

Références

- C. Hirt, A. Amsden : *Arbitrary Lagrange-Eulerian computing method for all flow speeds*, Journal of Computational Physics, 14, 3, 227–253 (1974)
- P. Woodward, P. Colella : *The numerical simulation of two-dimensional fluid flow with strong shocks.* Journal of Computational Physics ,54 , 115–173, (1984).
- G. Wellein, T. Zeiser, S. Donath and G. Hager : *On the Single Processor Performance of Simple Lattice Boltzmann Kernels.* Proc. ICMMES, 2004. Computers & Fluids 35, 910-919 (2006).
- S. Williams, A. Waterman and D. Patterson : *Roofline: an insightful visual performance model for multicore architectures.* Commun. ACM 52, 4, 65-76 (2009).

Références II

J. Treibig, G. Hager : *Introducing a performance model for bandwidth-limited loop kernels.* Parallel Processing and Applied Mathematics, Lecture Notes in Computer Science, vol. 6067, 615–624 (2010)

J. Treibig, G. Hager and G. Wellein : *LIKWID: A lightweight performance-oriented tool suite for x86 multicore environments.* Proceedings of PSTI2010, the First International Workshop on Parallel Software Tools and Tool Infrastructures, San Diego CA, September 13, (2010)

Intel : *Intel architecture code analyzer*, https://software.intel.com/en-us/articles/intel-architecture-code-analyzer/
(2011).

Références III

J. Treibig, G. Hager, H. G. Hofmann, J. Hornegger, and G. Wellein : *Pushing the limits for medical image reconstruction on recent standard multicore processors.* International Journal of High Performance Computing Applications 27(2), 162–177 (2013).

J.-M. Etancelin : *thèse : Couplage de modèles, algorithmes multi-échelles et calcul hybride*, (2014).

Merci de votre attention

Identification du peak spécifique d'un kernel

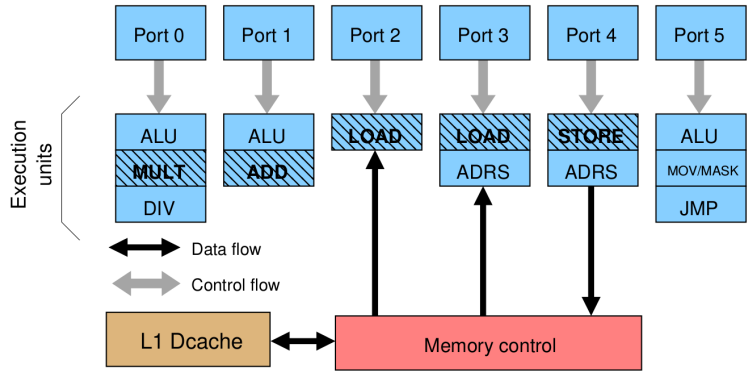


Figure : modèle bas niveau d'un cœur (unités d'exécution)

```
| Num Of |              Ports pressure in cycles               |    |
|  Uops  |  0  - DV  |  1  |  2  -  D  |  3  -  D  |  4  |  5  |    |
---------------------------------------------------------------------
|   1    |           |     | 1.0   1.0 |           |     |     |    | vmovsd xmm0, qword ptr [rip+0xf5e]
|   2    | 1.0       |     |           | 1.0   1.0 |     |     | CP | vmulsd xmm1, xmm0, qword ptr [rbx+r12*8]
|   2    |           | 1.0 | 1.0   1.0 |           |     |     | CP | vsubsd xmm2, xmm1, qword ptr [r15+r12*8]
|   2    | 1.0       |     |           | 1.0   1.0 |     |     | CP | vmulsd xmm3, xmm2, qword ptr [r13+r12*8]
|   2    |           |     | 0.5       | 0.5       | 1.0 |     | CP | vmovsd qword ptr [r14+r12*8], xmm3
|   1    |           | 0.2 |           |           |     | 0.9 |    | inc r12
|   2^   |           |     | 1.0   1.0 |           |     | 1.0 |    | cmp r12, qword ptr [rsp+0x8]
|   0F   |           |     |           |           |     |     |    | jl 0xffffffffffffffcd
|   1    |           |     |           | 1.0   1.0 |     |     | CP | mov r12d, dword ptr [rsp]
```

Figure : exemple d'utilisation de IACA pour un kernel
$d[i] = (1.5 * a[i] + b[i]) * c[i]$