DE LA RECHERCHE À L'INDUSTRIE

cea

www.cea.fr

Performance modeling of an industrial hydrocode on recent multicore processors

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

[1] CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235, Cachan, France
[2] Maison de la Simulation, USR 3441, CEA – CNRS – INRIA – Univ. Paris-Sud – Univ. de Versailles, F-91191, Gif-sur-Yvette, France
[3] CEA, DAM, DIF, F-91297 Arpajon, France
[4] CEA, DEN, DM2S, STMF 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

DE LA RECHERCHE À L'INDUSTRIE
cea

SimRace 2015

8^{th} december 2015

Outline

The goal

Build a solver for compressible fluid dynamics (Euler equations) which will run efficiently on current and future computers

How can we get this ?

Understand HPC properties of the reference numerical method

- Analyze and understand the current solver performance

Improve the current solver to its limits

- Understand current processor micro-architectures
- Optimize the solver implementation for current processors micro-architectures
- Estimate the maximal achievable performance for this solver

Going further

- Predict performance on other and future architectures
- Design a solver based on a more efficient algorithm

How can we get this ?

Understand HPC properties of the reference numerical method

- Analyze and understand the current solver performance

Improve the current solver to its limits

- Understand current processor micro-architectures
- Optimize the solver implementation for current processors micro-architectures
- Estimate the maximal achievable performance for this solver

Going further

- Predict performance on other and future architectures
- Design a solver based on a more efficient algorithm

$\Rightarrow$ Building performance model

Specificity of our solver (Lagrange-Remap methods)

1 - Lagrangian step

Compute evolution of hydrodynamic quantities on a mesh moving at material velocity

2 - Remap step

Remap / interpolation from distorted mesh to fixed initial one

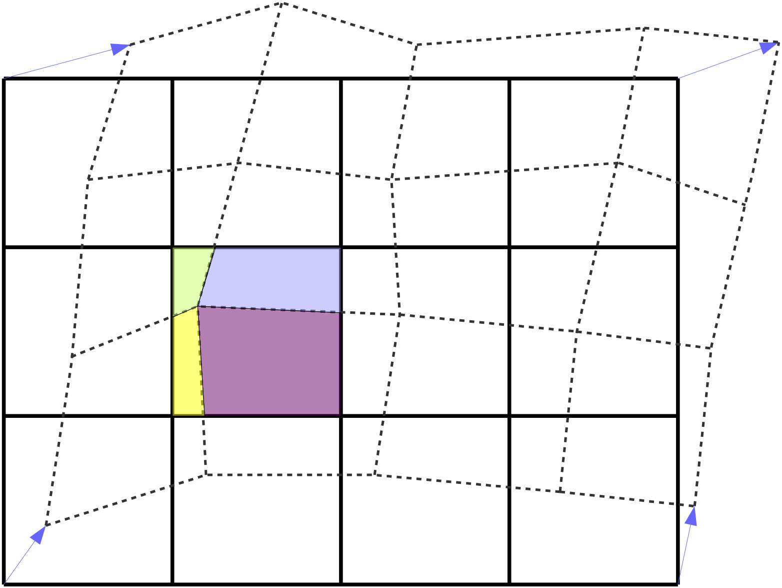


Figure : Sketch of Lagrange-Remap methods

Outline

What is performance modeling ?

Definition

Build and use simple abstraction to estimate execution time of a chosen algorithm on a given computing architecture
$\Rightarrow$ Understanding the performance behavior of a code

Interests

- Define how efficiently a computer is used (current usage vs machine peak)
- Predict / extrapolate performance behavior on others architectures
- Provide optimization hints

Performance modeling: principle

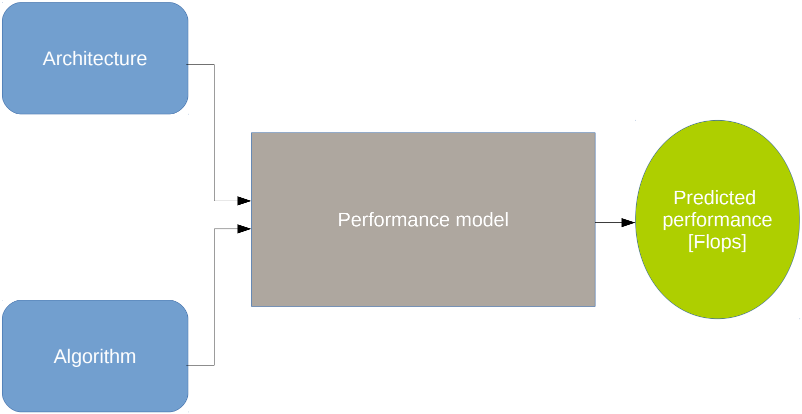


Figure : Input / output of a performance model

The Roofline model [Williams 2009]: metrics

Architecture

- **Bandwidth**:
  Data transfers rate from memory to computing units[GBytes/s]
- **Peak**:
  Absolute maximum performance of a computer[GFlops]

Algorithm

- **Arithmetic Intensity (AI)**:
  $AI := \frac{\text{number of operation}}{\text{quantity of data transfered}}$ [Flops/Byte]
- **Unbalance factor (instruction mix)**:
  Reduction performance factor due to not perfect matching between algorithm and architecture [%peak]

The Roofline model: definitions

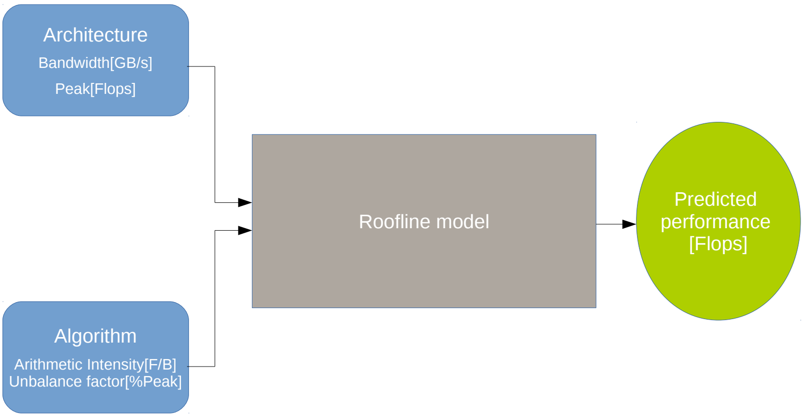
(ideal) Roofline model

$$\begin{aligned} \text{Perf(ideal)} &= f(Peak, bandwidth, AI) \\ &:= min(Peak, bandwidth \times AI) \end{aligned}$$

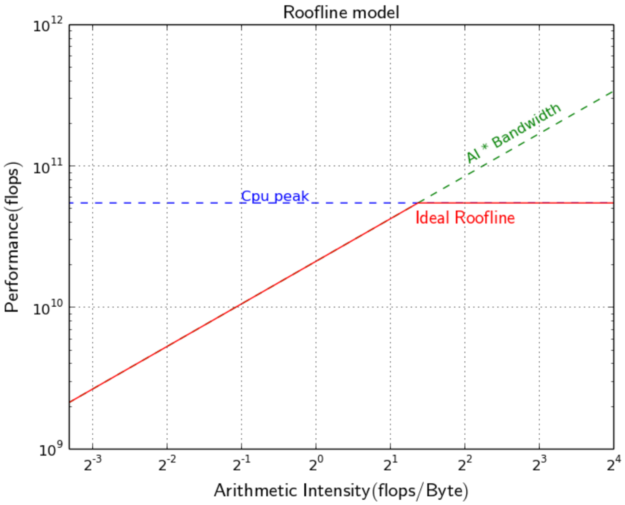
(effective) Roofline model

$$\begin{aligned} \text{Perf(effective)} &= f(\text{Peak}, \text{bandwidth}, \text{AI}, \text{unbalance factor}) \\ &:= min(\text{Peak} \times \text{unbalance factor}\ ,\ \text{Bwd} \times \text{AI}) \end{aligned}$$

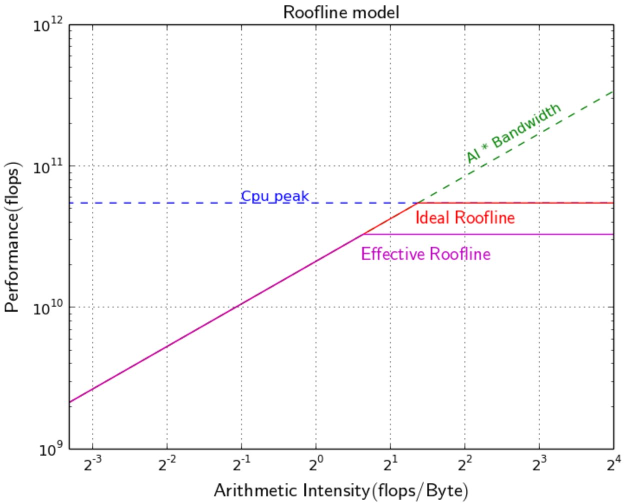
Roofline: principle



Roofline: graphical representation



Roofline: graphical representation



Outline

Dataflow diagram of the remap step

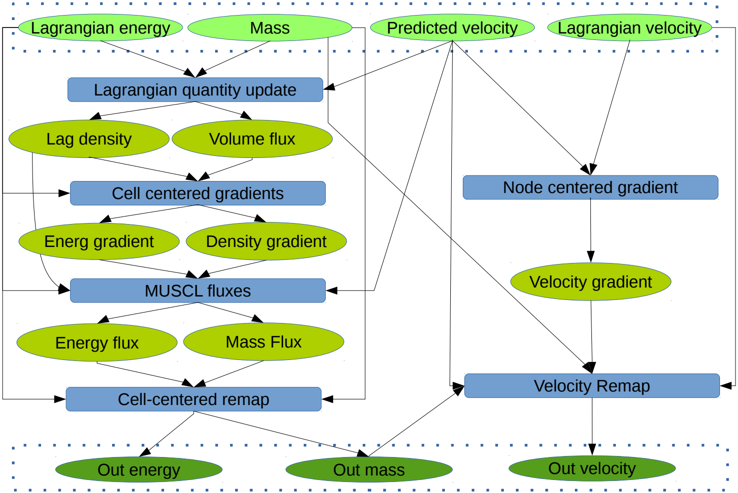
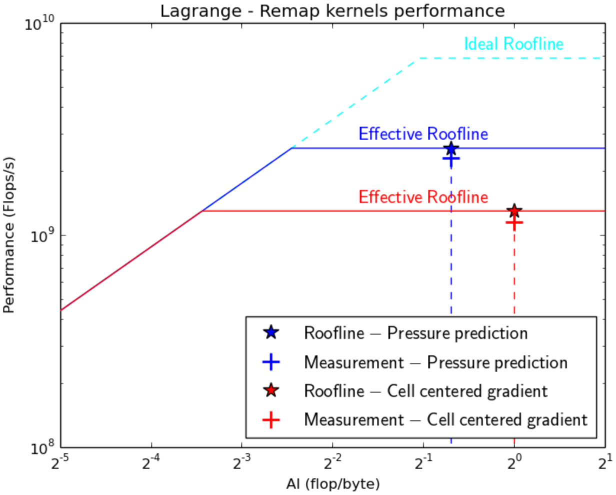


Figure : Dataflow diagram of the remap step

Roofline model vs measured performance

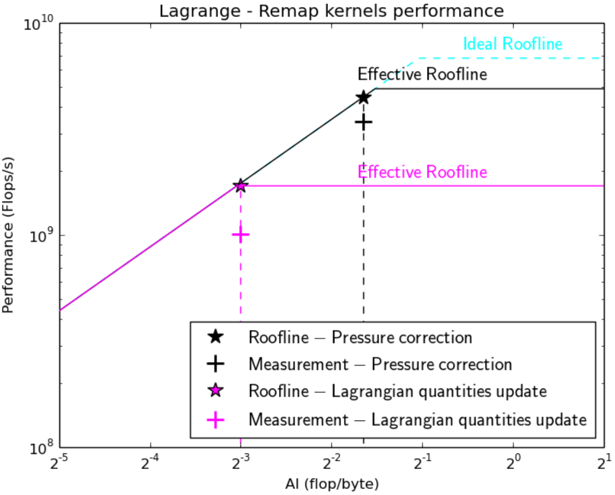


error

11%

12%

Roofline model vs measured performance



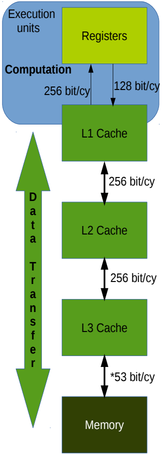
error
30%

70%

Outline

Execution Cache Memory model [Hager & Treibig 2010]

ECM (Execution Cache Memory model): refinement of the Roofline model by introducing data transfers through caches



Metrics: time [cycles / cache line update]

- Time for pure computation (assuming data are in L1 Cache)
- Times for data transfers through the different cache levels L1-L2, L2-L3, L3-Ram

Main hypotheses

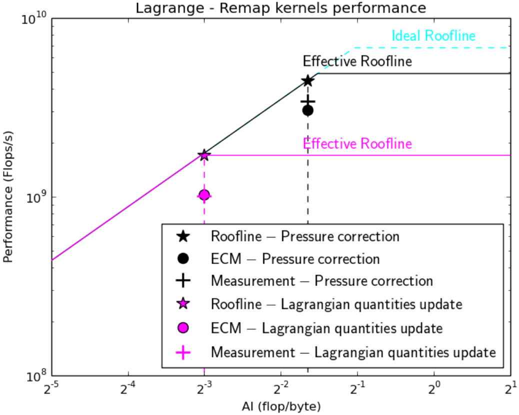
- Computations and data transfers may overlap
- Data transfers do not overlap

ECM model vs measured performance

Haswell AVX kernels	ECM	measurements	error
Lagrange kernels			
Pressure prediction	168	173	3%
Pressure correction	80	78	3%
Velocity update	89	65	8%
Remap kernels			
Lag. q. update	57	58	2%
Cell centered gradient	168	170	1%
MUSCL Flux	44	42	5%
Cell centered remap	76	65	17%
Node centered gradient	168	170	1%
Velocity remap	30	25	17%

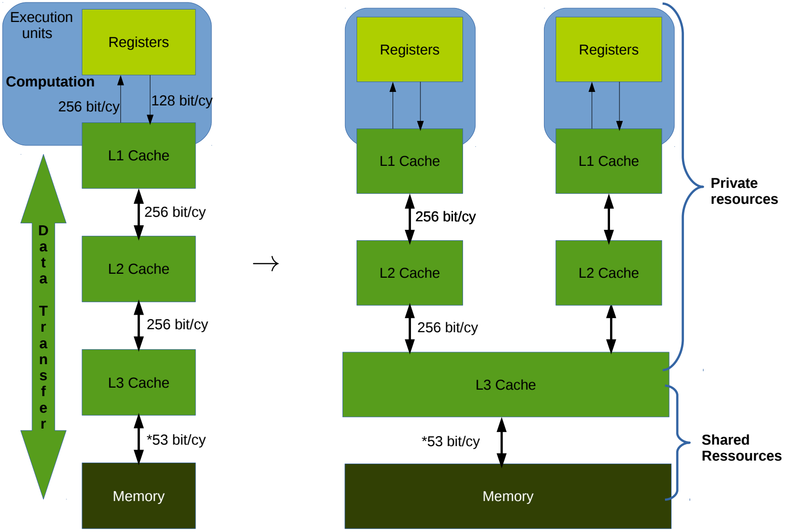
ECM model vs Roofline model vs measured performance

kernel name	time [cycles / cache line update]	
	prediction ECM	measurement
Pressure correction	80	78
Lagrangian q. update	57	58



modeling error	
Roofline	ECM
30%	3%
70%	2%

ECM: from one core to full node



ECM: multicore scalability

Kernel name	Speed up 8-cores SNB		Speed up 4-core HSW	
	Pred.	Meas.	Pred.	Meas.
Lagrange kernels				
Pressure prediction	8	7.45	4	3.5
Pressure correction	2.6	3.0	1.4	1.6
Velocity update	2.8	3.0	1.5	1.5
Remap kernels				
Lagrangian q. update	3.2	3.4	1.6	1.6
Cell centered gradient	8	7.9	4	3.9
MUSCL fluxes	2.3	2.6	1.3	1.4
Cell centered remap	2.5	2.7	1.2	1.6
Node centered gradient	8	7.9	4	3.9
Velocity remap	2.3	2.7	1.5	1.5

Conclusions and on going works

Performance modeling

- Very useful analysis tools for understanding code performance behavior
- Provide qualitative / quantitative information depending on the used model

Highlighting some bottlenecks in the current solver

- Variable location (node and cell-centered)
- Multi dimensional remap in multiple steps
- Complex geometrical features

On going works

- Building a new efficient numerical scheme (see F. De Vuyst talk, Thursday 10^{th})
- Performance modeling of this new scheme

Thanks for your attention!