

CALMIP : EVOLUTION ARCHITECTURE PROCESSEUR

Mésocentre CALMIP – Nicolas Renon

RCTF 2018 11/10/2018



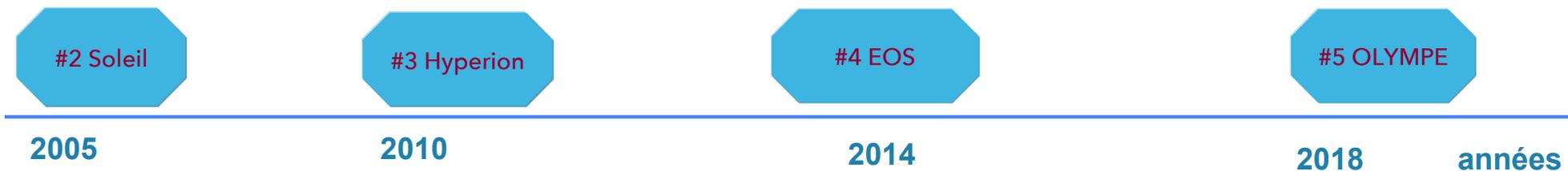
CALMIP (UMS 3667)
Espace Clément Ader
www.calmip.univ-toulouse.fr



AGENDA

- ▶ CALMIP : 15 ans d'Evolution Processus
- ▶ Quels Impacts sur les codes de calcul ?
- ▶ Vectorisation, actuellement ...
- ▶ Intensité Arithmétiques et Roofline Model
- ▶ Tendances Futures

GÉNÉRATIONS DE SUPERCALCULATEURS ET DE PROCESSEURS



Processeur :
Mono-core

Processeur :
Multi-core



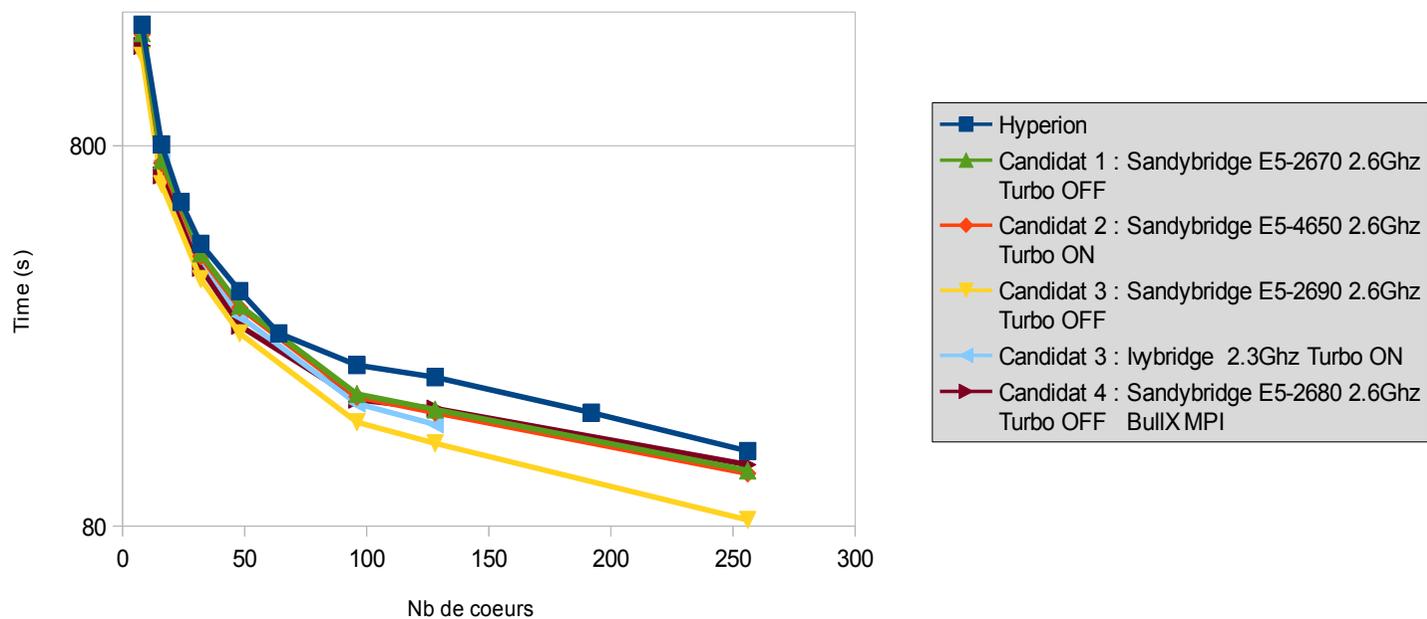
BUFFET GRATUIT!!!

LE BUFFET EST PLUS TELLEMENT GRATUIT !!!

CAMPAGNE DE BENCH 2012-2013



MESH 256x256x240



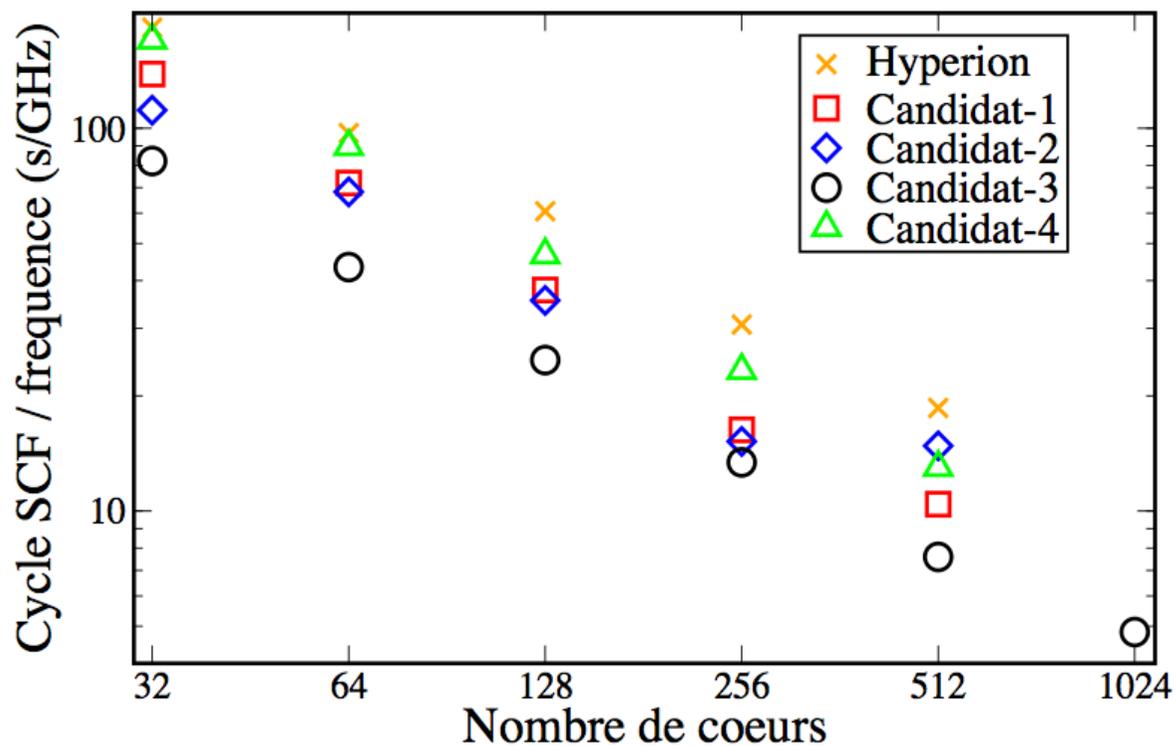
Mécanique des Fluides - Code Jadim, Annaïg Pedrono, IMFT - Code Neptune_CFD, Hervé Neu, IMFT

« IVYBRIDGE (Architecture EOS) 10% à 20% + rapide vs. NEHALEM (Architecture HYPERION) pour le même nombre de core »

CAMPAGNE DE BENCH 2012-2013



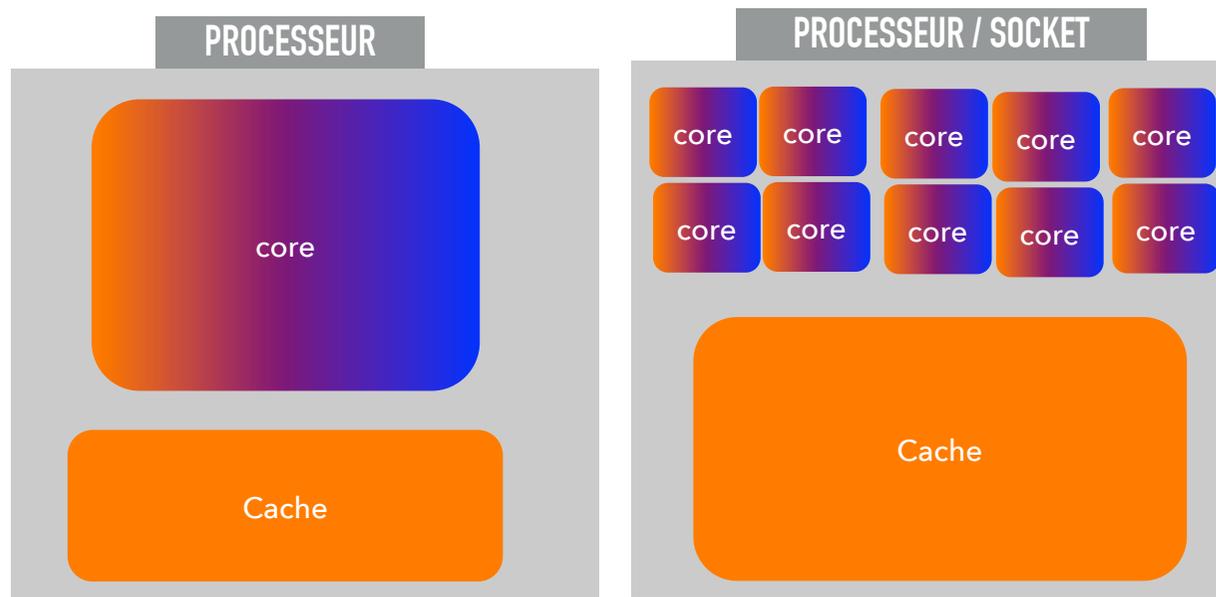
VASP Test-1



Code Vasp - Porteurs de bench : Iann Gerber (IRSAMC-LPCNO), Hao Tang (CEMES)

« IVYBRIDGE (Architecture EOS) presque 2 fois + rapide vs. NEHALEM (Architecture HYPERION) pour le même nombre de core »

ARCHITECTURE PROCESSEUR : MULTI-CORE



Gravure 180 nm
(ITANIUM) SOLEIL

6 Gflop/s

X40

Gravure 22 nm
(IVYBRIDGE) EOS

220 Gflop/s

GRAVER plus fin \Rightarrow plus de transistor \Rightarrow plus de core \Rightarrow plus de capacité de calcul
(enveloppe énergétique contrôlée)

SCALAIRE VS. VECTORIEL (SIMD)

Core scalaire



1 CPU cycle

Core vectoriel



Courtesy of A. Scemama

Avec cette vectorisation : Flop/s x 4 vs. Scalaire

ARCHITECTURE PROCESSEUR : APPORT VECTORISATION



	core	core	core
	HYPERION Intel® NEHALEM	EOS Intel® IVYBRIDGE	OLYMPE Intel® SKYLAKE
Frequency Ghz	2,80	2,80	2,3
# compute unit	2	2	2
vectorisation(DP)	2	4	8
	128bit	256 bit	512 bit
1 core	11 Gflop/s	22Gflop/s	73* Gflop/s

* FMA : Fuse Mult-Add

Un gain par core grâce à la vectorisation
Tous les codes n'en tirent pas partie

NOTION D'INTENSITÉ ARITHMÉTIQUE

► Intensité Arithmétique = (# Flop) / (# Bytes)

```
do i=1,n  
  a_vec(i)=b_vec(i)*c_vec(i)+d_vec(i)*e_vec(i)  
enddo
```

Programme n° 1 :

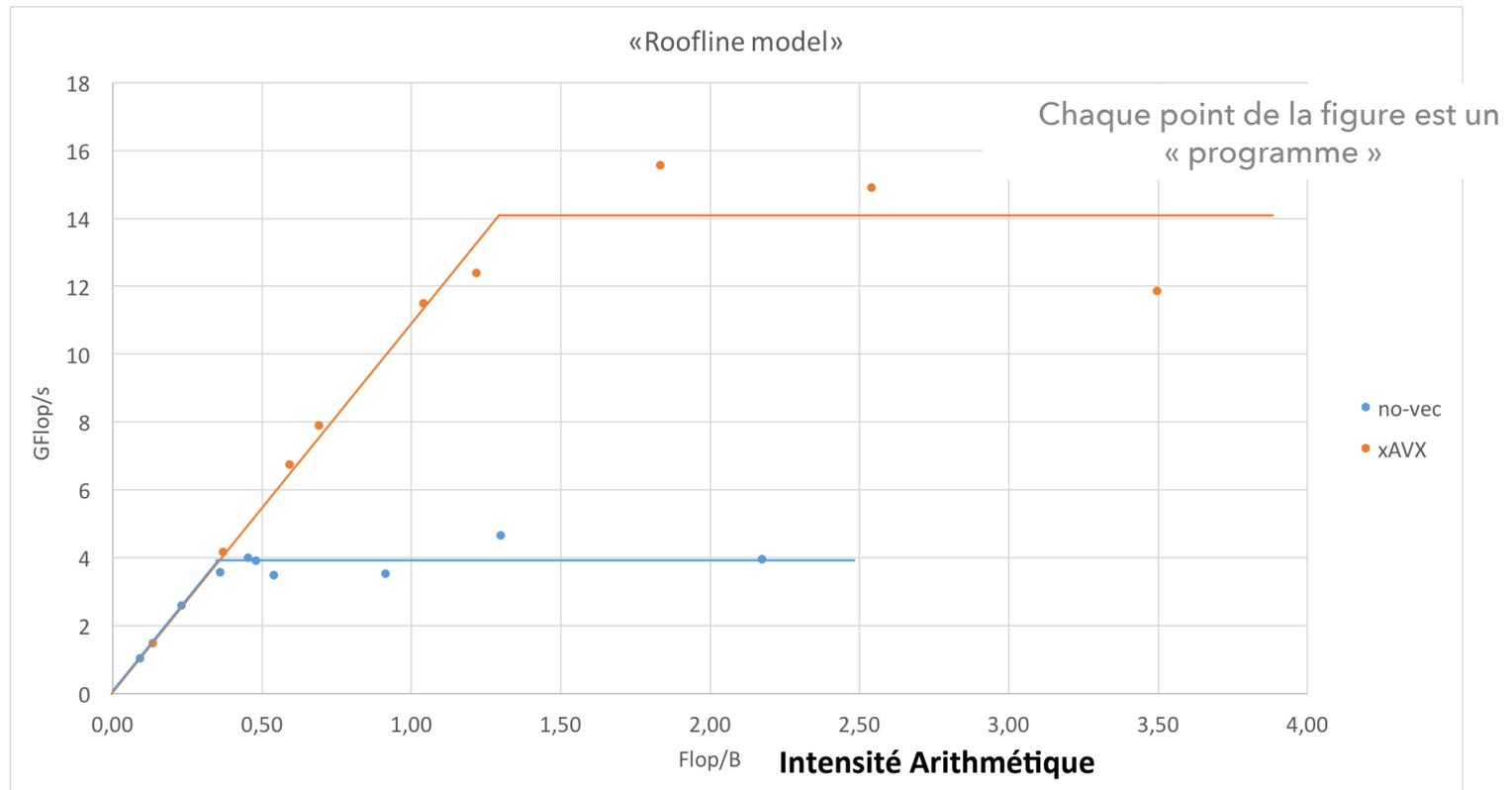
$(3 \text{ Flop}) / (5 * (8 \text{ B})) = 0,075 \text{ Flop/B}$

M. Duval, Equipe Support CALMIP

<https://www.calmip.univ-toulouse.fr/spip.php?article552>



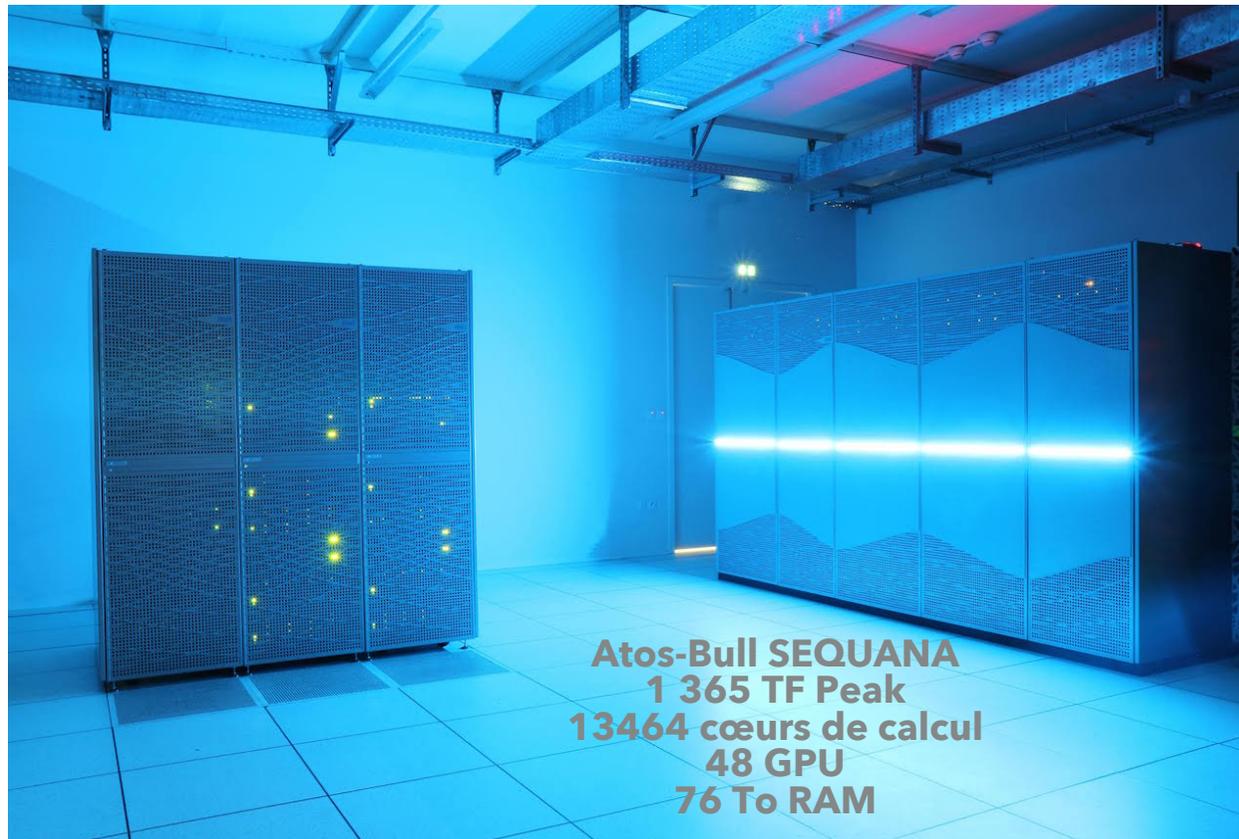
Performance



Plus l'intensité arithmétique augmente, plus la performance augmente

- ▶ Pas de gain gratuits, ni magiques !! Perf des codes dépend de leur adéquation à l'architecture !
- ▶ Processeurs :
 - ▶ + de vectorisation (intel : 512 bit), intensité arithmétique élevée (#1@TOP500 (nov 2017) : IA = 22flop/byte)
 - ▶ La fréquence diminue
- ▶ « Dura (physicae) Lex Sed Lex » : fin de la « Loi de Moore » ?
 - ▶ Gravure : 22 nm = OK, 14 nm =~ OK,
 - ▶ Futur : 10 nm ? 7 nm ? 5 nm ?
- ▶ GPU : Supercalculateur Olympe CPU et GPU

OLYMPE



Atos-Bull SEQUANA
1 365 TF Peak
13464 cœurs de calcul
48 GPU
76 To RAM

TFLOP/S GPU VOLTA = 6 × TFLOP/S CPU SKL

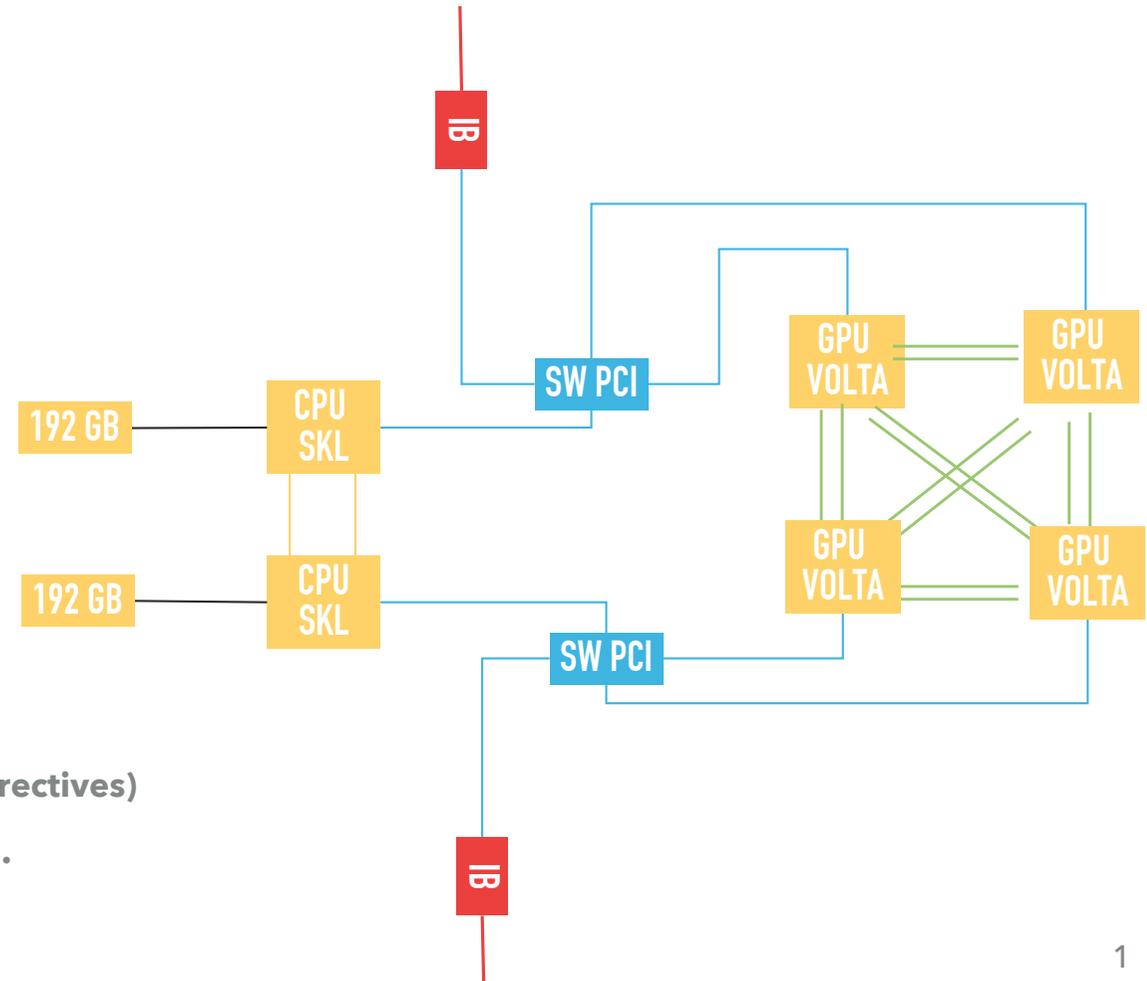
UPI 10 GT/S

PCI GEN3 16X 15.8 GB/S

NV-LINK 2.0 25 GB/S

IB EDR 4X 12 GB/S

128 GB/S



- Massive fine grain parallelism
- Programming : CUDA / OpenMP / OpenACC (Directives)
- Data access : a performance issue
- Code GPU Ready : Amber, Gromacs, (~) VASP, ...