DE LA RECHERCHE À L'INDUSTRIE

EVOLUTION DES MACHINES ET IMPACT SUR LES MÉTHODES NUMÉRIQUES

GDR Mascot num | D Bouche et G Colin de Verdière

16/11/2015

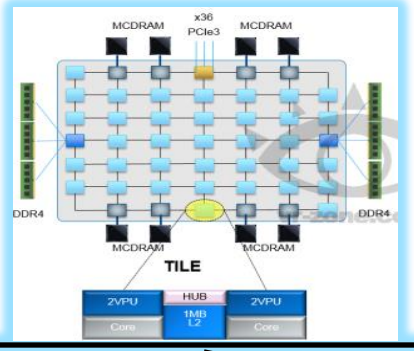
Sommaire

Contexte

Ce qu’on constate sur les machines de puissance

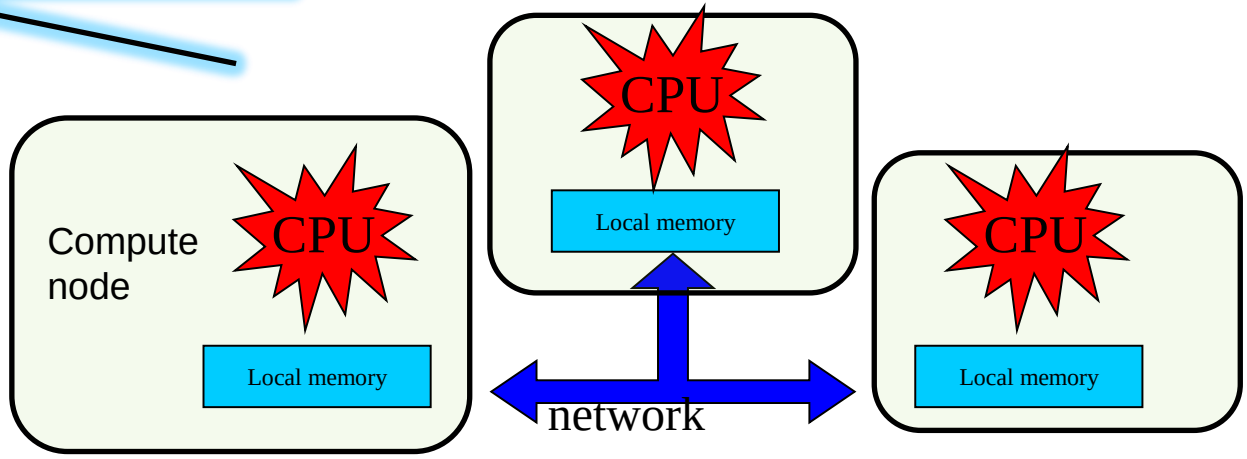
Des architectures regroupant un grand nombre de noeuds complexes et hétérogènes

- des noeuds de plus en plus complexes



Des noeuds de plus en plus nombreux
(>= 5000)

- Architecture hybride
- Vectorisation
- Multithreading
- Grand nombre de coeurs
- Effets NUMA
- Mémoire par coeur réduite



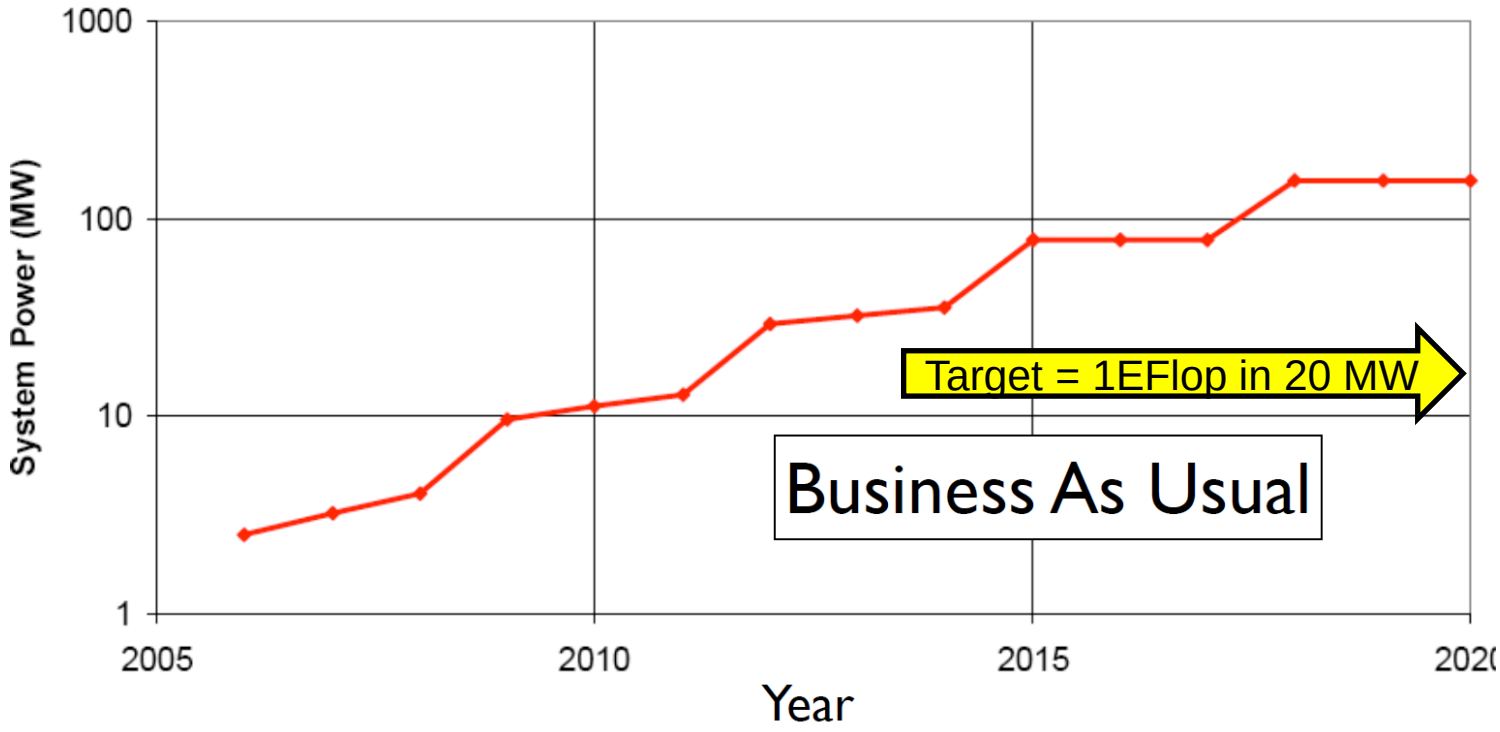
Le powerwall

L’électricité, c’est cher…

1MW=1 Meuro/an

Tricastin © AFP/Fred Dufour

Premier défi: la puissance consommée



Credit K.J. Roche, Jul 20**11**, U. of Michigan

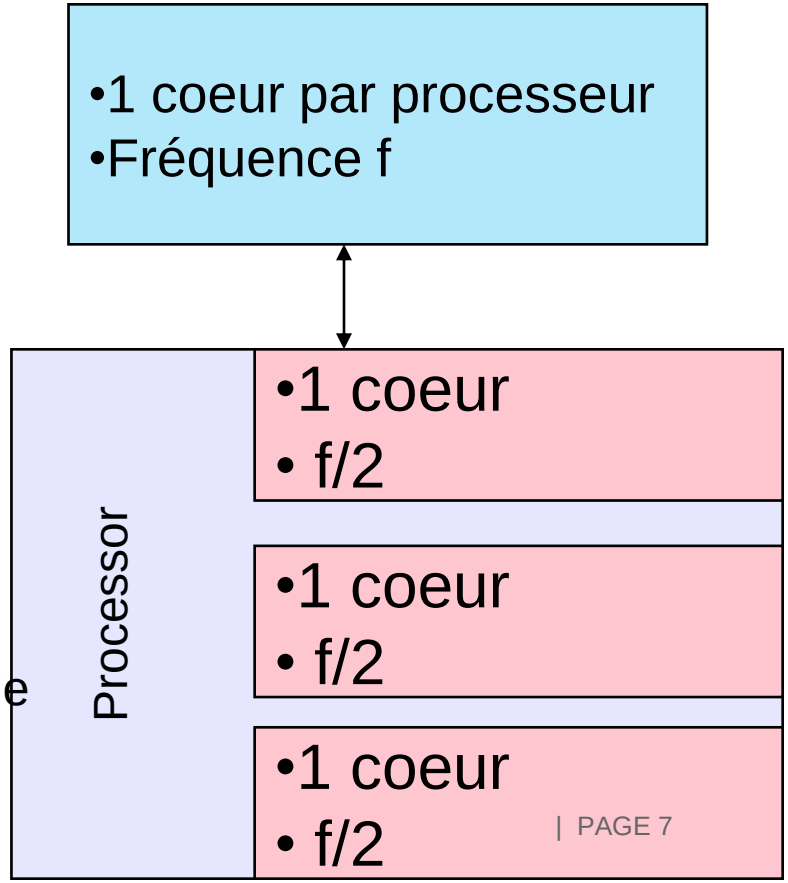
Contrainte : puissance consommée: $P = P_d + P_s$

- $P_d = C_e \times f \times V^2$ **dynamique**
- $P_s = V \times I_f$ **statique**

- I_f : courant de fuite
- C_e : capacité
- V : voltage(fonction croissante de f)
- F : fréquence

- **Pour limiter P, il faut**
  - **Réduire le voltage**
    - **ex: Pentium 4 = 1.7V; Nehalem = 1.247V**
  - **Réduire la fréquence f**
    - **Nehalem = 2.8GHz, GPU = 1.1GHz**
- **Capacité calcul en $N_c\ N_{op}\ f$**
  - **$N_c\ N_{op}$ doit augmenter si f baisse**
- **Conséquence**
  - **Parallélisme**

N_c nombre de cœurs
N_{op} nombre d'op/cycle

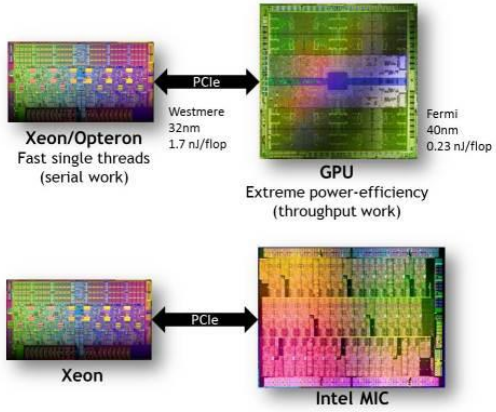


Plus de performance = plus de coeurs

- Les solutions sont soumises aux contraintes du marché
  - Ordinateurs hybrides /classiques
  - Top500 : 8 hybrides dans le top 20 #1 et #2 sont hybrides

- Road Runner (le précurseur)
  - Processeur de console de jeu

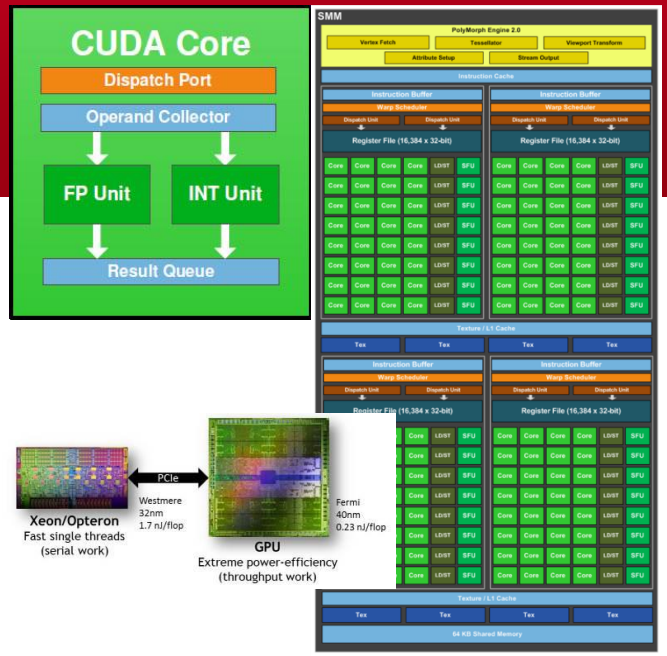
- La piste « GPU » NVIDIA, AMD
  - Monde du jeu et du graphique
  - Un grand nombre de coeurs simples



- La piste « Manycore » INTEL
  - Monde du PC
    - Capitalise un écosystème logiciel existant
  - Un nombre raisonnable de coeurs de puissance intermédiaire

- Demain, processeurs embarqués à faible consommation
  - Monde du téléphone portable, des tablettes…

La piste “GPU”

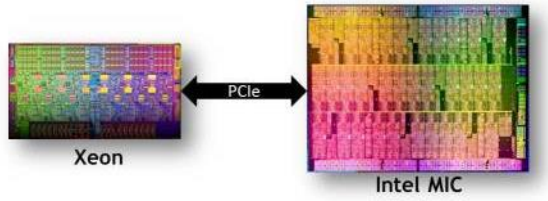


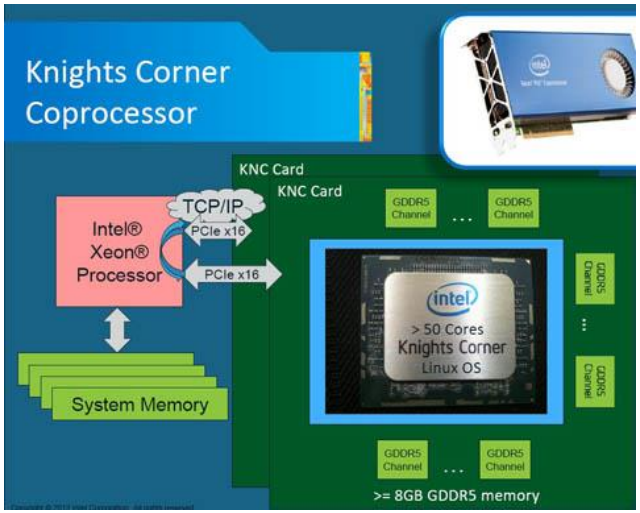
- NVIDIA & AMD
- travail sur les pixels : parallélisme massif
  - S’appuie sur des programmes à grand nombre de threads
  - Le hardware introduit des contraintes sur le style de programmation
  - Peut être vue comme une machine vectorielle (data parallel) : le processeur exécute la même opération sur un ensemble de données
- Performance (potentielle…) élevée
  - Programmation spéciale
  - Demande un CPU et un lien externe (PCIExpress, NVlink)
    - importance de la localité des données
    - coût des mouvements de données

The ROMEO machine @ URCA
#184 Top500 06/14(384TFlop/s)
#6 Green500
NVIDIA K20X

La piste “Manycore”

- Suivie par Intel
  - Garde la compatibilité avec architecture X86
  - Revendique la simplicité de programmation
  - Considéré par Intel comme “la” route vers l’Exascale
    - Evolution de tout l’écosystème logiciel à prévoir (quand même…)

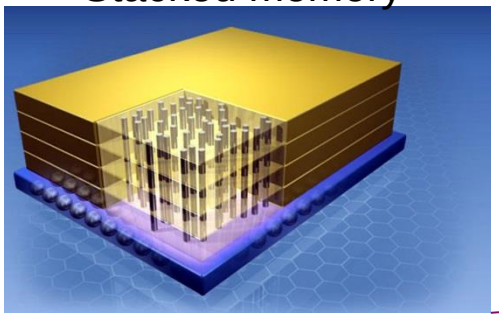




Tianhe-2, NUDT, China
#1 Top500 06/14
33,86 petaflops
32000 Ivy-Bridge
48000 Xeon Phi (KNC)
Bientôt 100 petaflops avec les « China accelerators »

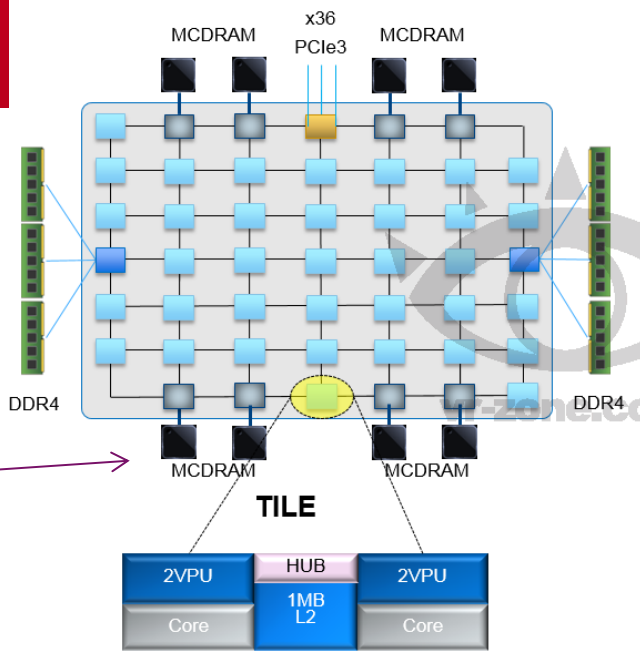
Xeon Phi Manycore KNL (2015)

Stacked memory



HMC by Micron

Knights Landing Processor Architecture



Up to 72 Intel Architecture cores based on Silvermont (Intel® Atom processor)

- Four threads/core
- Two 512b vector units/core
- Up to 3x single thread performance improvement over KNC generation

Full Intel® Xeon processor ISA compatibility through AVX-512 (except TSX)

6 channels of DDR4 2400 MHz -up to 384GB

36 lanes PCI Express* Gen 3

8/16GB of high-bandwidth on-package MCDRAM memory >500GB/sec

200W TDP

- **Architecture**
  - 72 coeurs + 4 threads / coeur
  - **Multithreading nécessaire**
- **2 unités vectorielles 512b par coeur**
  - L'essentiel de la puissance de calcul du KNL
  - **Programmation vectorielle obligatoire**
- **Mémoire stackée [HMC] (B/W ↑, latence ≡)**
  - Impact sur les codes (placement en mémoire)
- **Effets NUMA internes**

N_c nombre de cœurs : 72
N_{op} nombre d'op/cycle : 32

72 x 32 x1,3 GHz : 3 TFlops

Le retour du vectoriel

```
do i = 1, n
   a(i) = b(i) + c(i)
enddo
```

- Seule manière de booster la puissance de calcul au niveau d'un cœur : l'énergie nécessaire pour une opération dépend peu de la taille des registres
- Déjà utilisé sur le Cray1

$$\begin{pmatrix} a_1 \\ \dots \\ a_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \dots \\ b_n \end{pmatrix} + \begin{pmatrix} c_1 \\ \dots \\ c_n \end{pmatrix}$$

- Apparaît sur les processeurs classiques il y a quelques années
- Registres d'abord de taille modeste : 128 bits (2 opérations DP)
- Puis la taille des registres augmente : 256, puis 512 bits (KNL, mais aussi futures génération de processeurs classiques
- Les 2 registres réalisent chacun 8 opérations simultanées fusionnées (fused multiply add : d=a.b +c), comptant pour 2 opérations : 2x8x2=32
- Programmation spécifique

En un cycle d'horloge (1 ns), on fait peu de chemin…

Second défi : les mouvements de données

C=2.997925 10^8 m/s

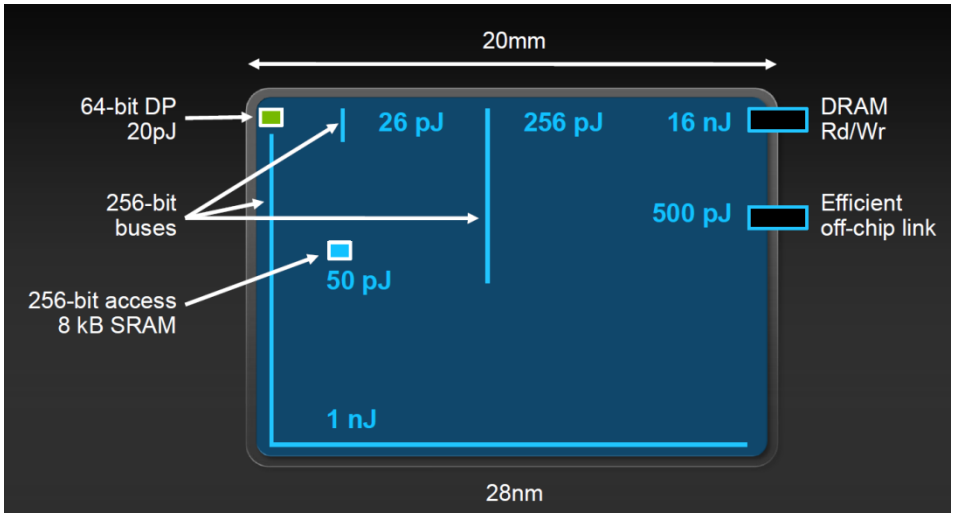
1ns ⇔ 30cm !

Déplacer les données coûte du temps et de l'énergie

- Calculer peut être plus économique que déplacer des données
- Hierarchies Mémoire et coûts associés

NUMA

L1$	small size (kB), small latency (~1cy) , high B/W
L2$	medium size (kB), medium latency (~14cy), high B/W
L3$	larger size (MB), high latency (~40 cy)
- MCDRAM (KNL)	very high latency (~140 cy), focus on bandwidth
- DDR4	very high latency (~140 cy), focus on storage size



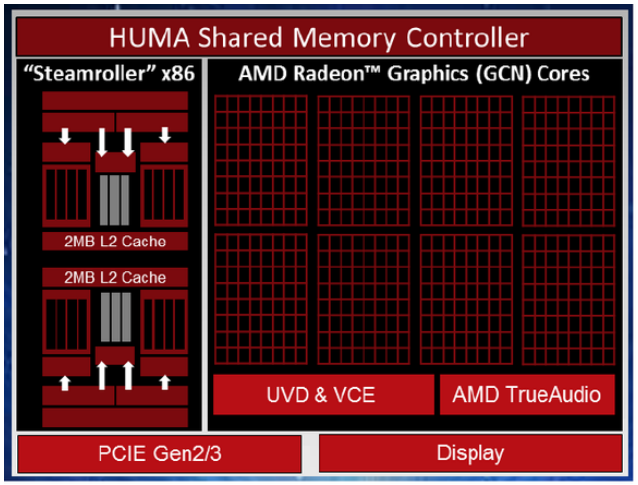
Credit Bill Dally, NVIDIA, SC10

architecture Fusion AMD
architecture Tegra NVIDIA

- Mettre le GPU au plus près du CPUs
  - Plus de communications via le PCI-Ex
  - Gestion des ressources plus facile pour les développeurs

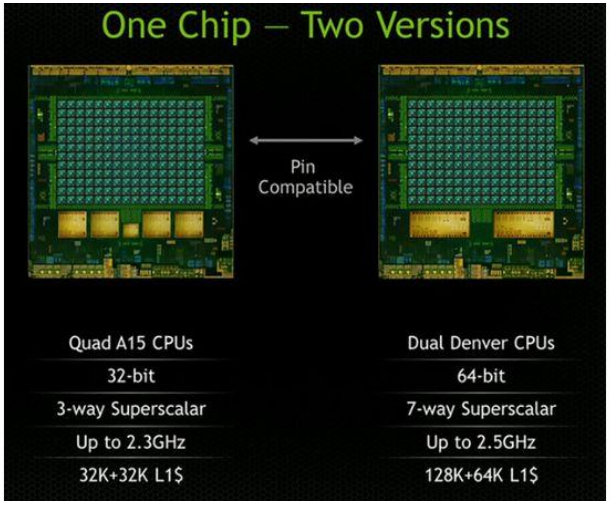
- Contraintes
  - modification de l'O/S
  - Un ecosystème logiciel pour ces architectures
    - Quels standards ?
      - AMD : OpenCL, OpenMP 4.0
      - NVIDIA : CUDA, OpenACC

- Pas encore applicable au HPC
  - marché = laptops, tablettes, smartphones
  - évolution rapide : à suivre…



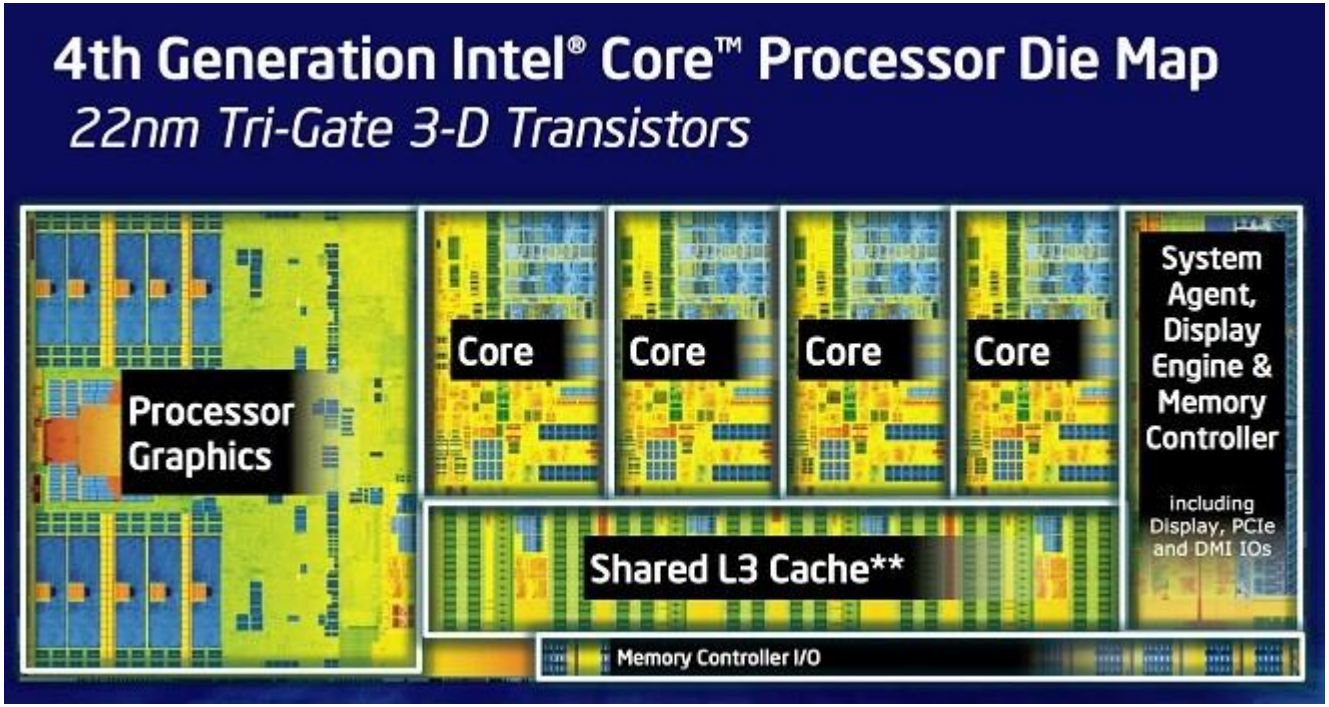
AMD Kaveri

NVIDIA Tegra k1



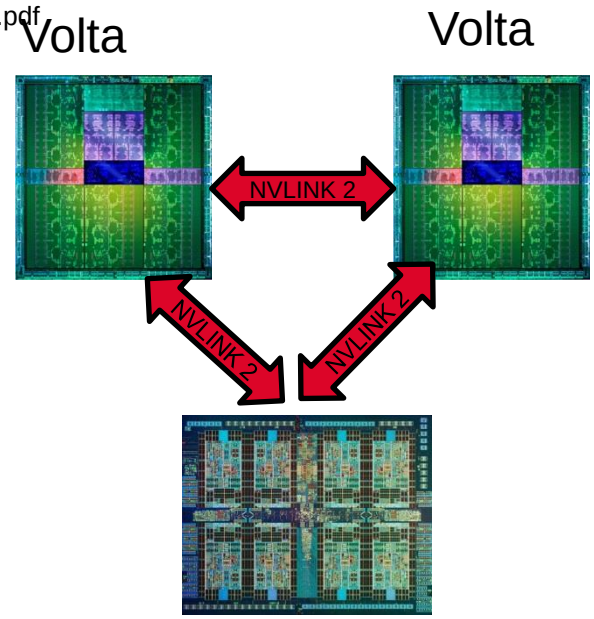
Remarque : chez Intel, les processeursrs classiques ont déjà ce type d'architecture

- Processseur Intel Haswell



Un lien plus rapide : le NVLINK

- Machine Coral : GPU + IBM Power 9
  - IBM Power 9 + NVLINK + Volta
    - http://info.nvidianews.com/rs/nvidia/images/Coral%20White%20Paper%20Final-3-2.pdf

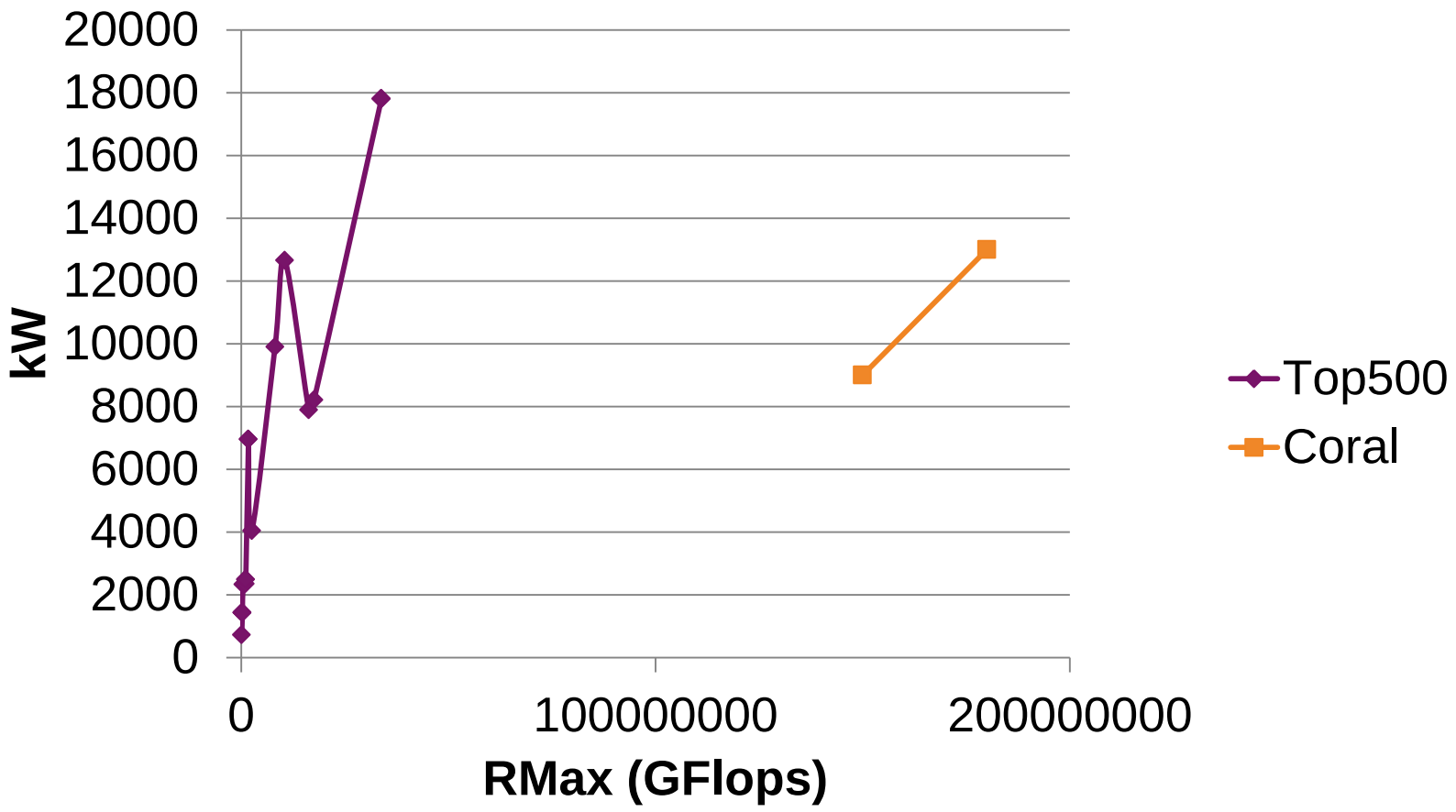


- NVLINK
  - PCI-Express Gen3 12 GB/s
  - Nvlink 1 64 GB/s
    - Disponible sur génération Pascal de GPU
  - Nvlink 2 vitesse au moins doublée

Coral : machines de classe 100 Pflops précurseurs de l'exascale

Coral : un bond en efficacité énergétique

- Coral vs top 500



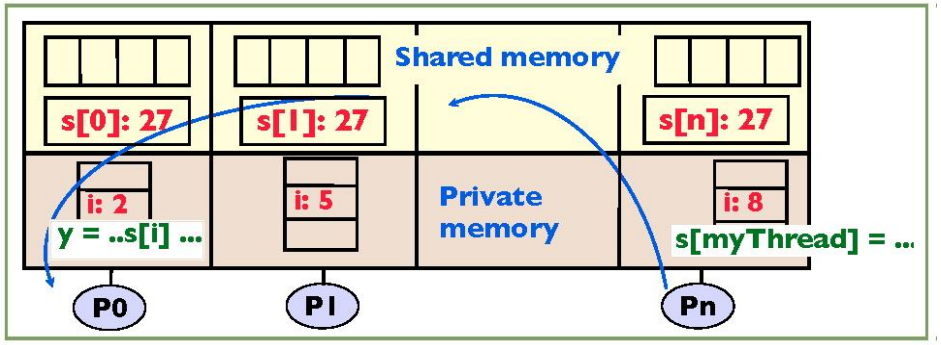
Les machines Coral

LLNL & ORNL	ANL
• IBM + NVIDIA	• INTEL + CRAY
• Power 9 + Volta ▪ NVLINK-2	• KNH (successeur du KNL)
• Mellanox Infiniband	• STL2
• ~150 PFlops • ~8 MW	• ~180 Pflops • 13 MW
• MPI + OpenMP 4	• MPI + OpenMP 4

Evolutions logicielles

Langages de programmation 1/2

- Fortran / C++
  - Déclin du FORTRAN ?
    - Encore une activité Open Source significative
  - Des évolutions intéressantes du C++ à venir
    - Meilleure intégration des threads
- PGAS
  - Concept intéressant encore loin de la production



Partitioned Global Address Space

langages de programmation 2/2

- Le modèle MPI +X est (encore) le future
  - question : quel X ?

- Options non standard
  - CUDA
    - Intéressant pour apprendre le parallélisme massif
  - OpenACC
    - Un bon labo pour OpenMP 4.x

- Options standard
  - OpenMP 4
    - Le meilleur pari pour le futur
      - Poussé par le programme Coral
  - OpenCL
    - Largement disponible

Méthodes numériques

Simulations numeriques: opportunités

- Le boost sur la puissance de calcul ouvre des opportunités :
- Simulations numériques “Classiques”
  - Des simulations d’écoulements bien résolues : $10^{9\text{-}11}$ elements aujourd’hui, $10^{12\text{-}14}$ sur machines exascale
  - resolution partielle de la malédiction de la dimension: simulations gyrocinetiques 5D
  - Prediction de la structure et des interactions entre molecules
  - structure électronique des materiaux, y compris avec electrons corrélés
- Couplage HPC / image-vision
  - structure des macromolécules biologiques par microscopie électronique et atomique
  - simulations Monte Carlo pour applications en radiothérapie
- Quantification des incertitudes et optimisation non-convexe

Simulations numériques: Constraintes pour obtenir des performances raisonnables

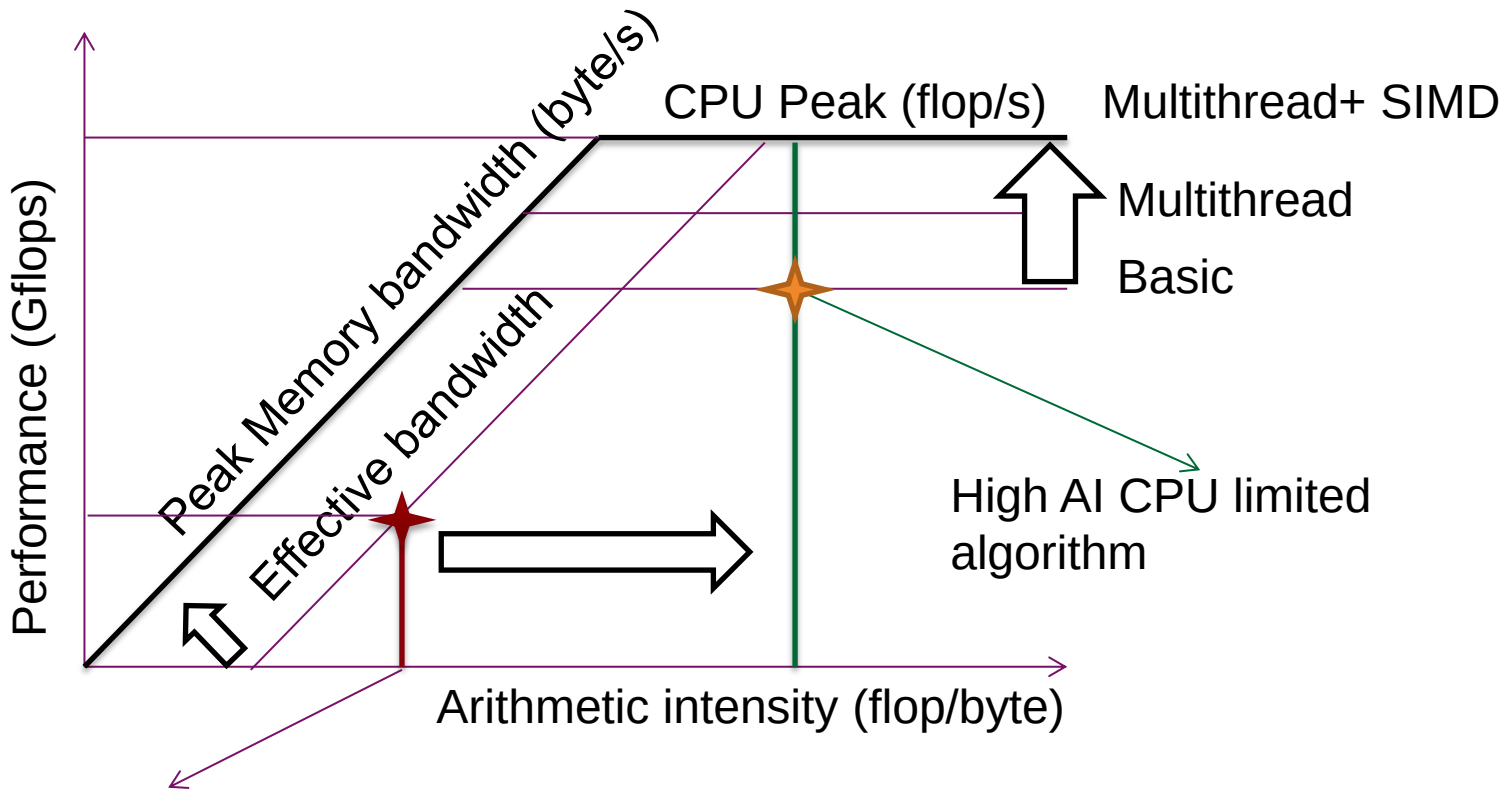
- Parallelism 3 niveaux
  - Entre nœuds de calcul : travail à répartir équitablement entre les noeuds
  - Dans les noeuds : multi ou manycore
  - Dans les coeurs : multithreading, unités SIMD (vectorielles)

- Négliger un de ces niveaux : mauvaises performances
  - Amdhal (scaling fort) et Gustavson (scaling faible) : même une petite fraction de code non parallèle limite drastiquement l'efficacité sur machines massivement parallèles
  - Les unités vectorielles sur Intel KNL réalisent 32 opérations simultanées: mal les utiliser limite l'efficacité à 1/32 du nominal…

- La bande passante mémoire n'a pas crû autant que la puissance de calcul
  - Mouvements de données plus coûteux que les calculs
  - Méthodes à faible intensité arithmétique limité par la bande passante : leur performance peut être une faible fraction du CPU crête
  - Problématiques d'optimisation de performance illustrées par le modèle roofline

Performance estimation and optimization

- Roofline : visual estimation of algorithm's performance

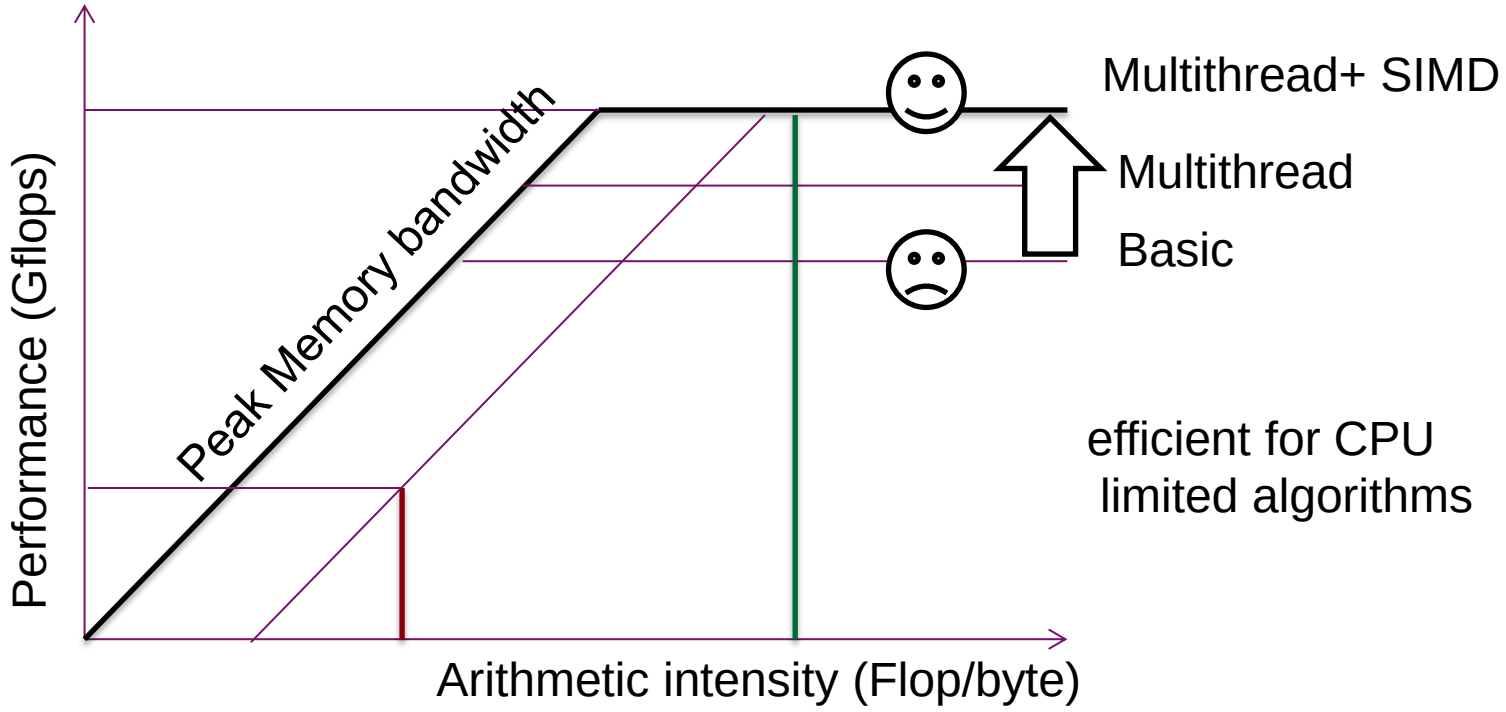


Low AI BW limited algorithm

How to get closer to peak CPU ?

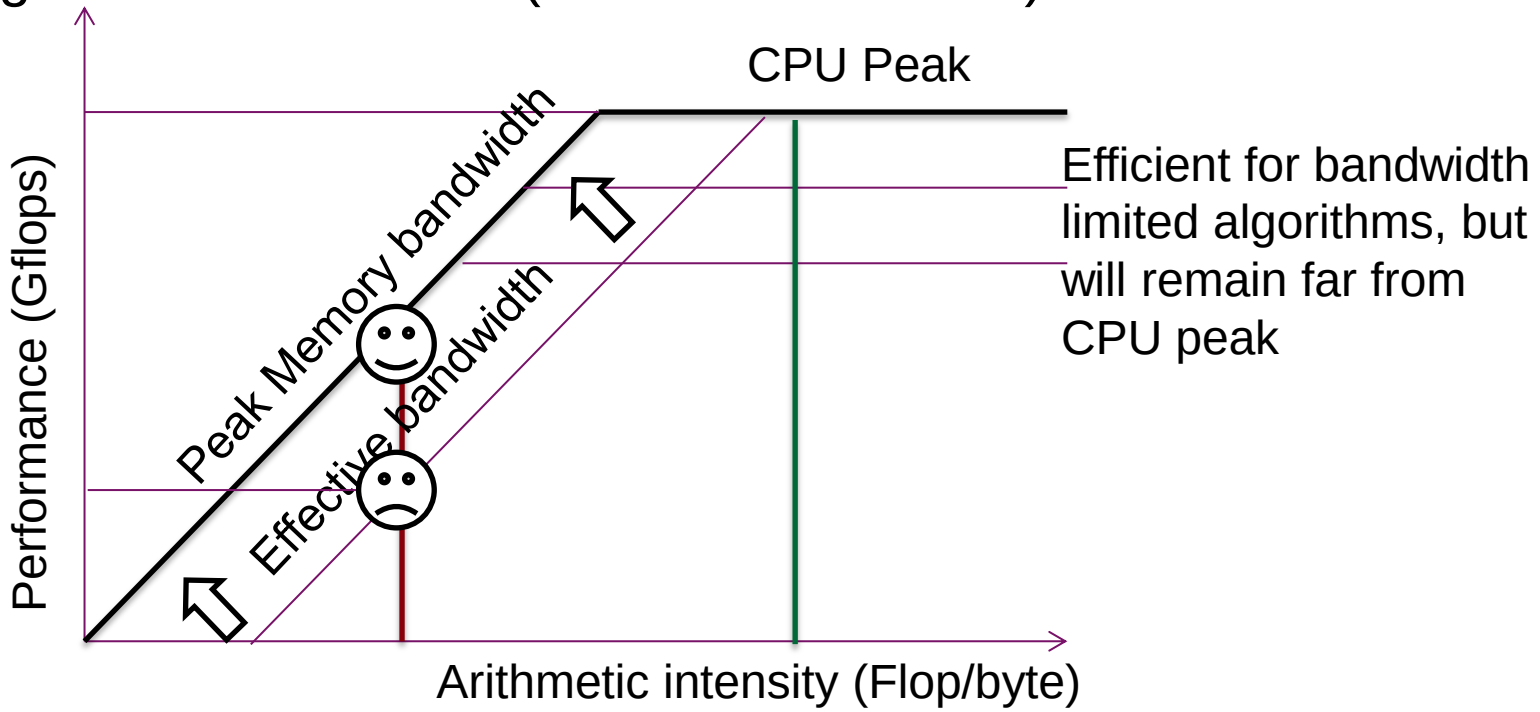
peak cpu quest : rise the ceiling

- Multithreading + efficient use of SIMD units (vectorisation) can enhance efficiency by several orders of magnitude (10-100)
- Vectorisation easier for regular data structures : finite differences, or structured meshes



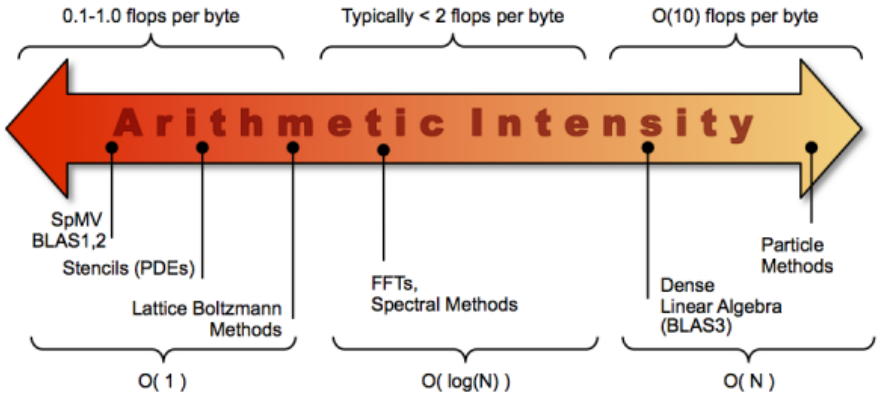
peak cpu quest : push the ROOF

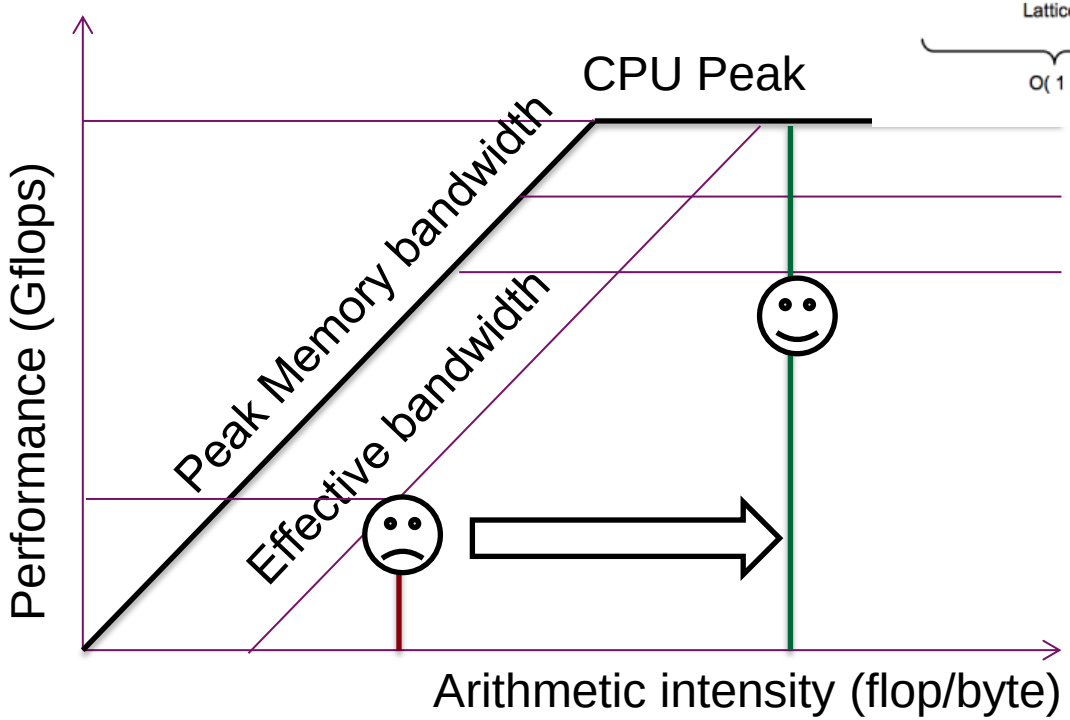
- Bandwidth can be reduced by parasitic memory traffic generated by cache misses, superfluous cache write allocations, etc.
- Regular access to data more efficient : again favoured by regular data structure (structured meshes)



peak cpu quest : move away from the ROOF

- Present and future machines :
- low byte/flop ratio
- Use high flop/byte algorithms



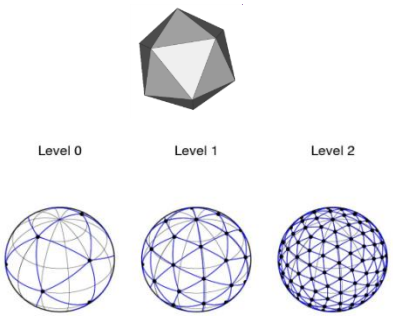


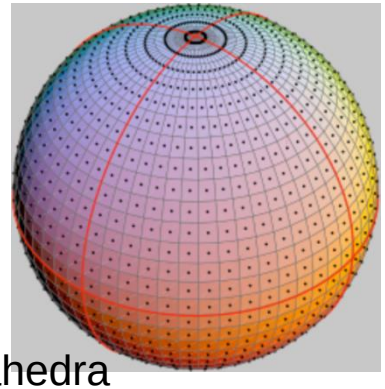
Example : Higher order stencils have higher arithmetic intensity

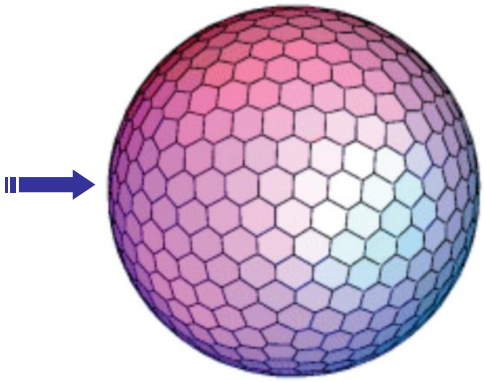
Massive parallelism

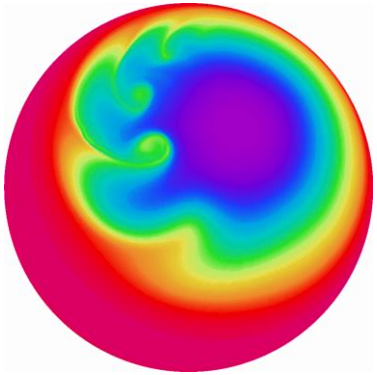
- Some algorithms are naturaly suited to parallelism
  - Monte Carlo methods : widely used for particle transport
  - Particle swarms : efficient in non convex optimization
  - Regular meshes : simpler to scatter between nodes
- With the advent of parallel machines, this requirement became more important than operation count
- Others require more work
  - Fourier transforms
  - Plane wave methods for electronic structure computations
- But it does not mean that they are ruled out…

DYNAMICO : A new atmospheric dynamical core (T. Dubos LMD/X, Y. Meurdesoif CEA/LSCE)









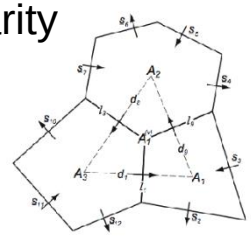
- Tessellation of icosahedra
- Hexagonal mesh + 12 pentagons
  - Removing mesh ...arity

$$\frac{\partial m}{\partial t} + \frac{\partial W}{\partial \eta} + \nabla_\eta \cdot (\overline{m}^h u) \quad m = -\frac{1}{g}\frac{\partial p}{\partial \eta}$$

$$\frac{\partial mq}{\partial t} + \frac{\partial}{\partial \eta}(W\overline{q}^v) + \nabla_\eta \cdot (U\overline{q}^h) = S_q$$

$$\frac{\partial \Phi}{\partial \eta} + g\frac{m}{\overline{\rho}^v} = 0$$

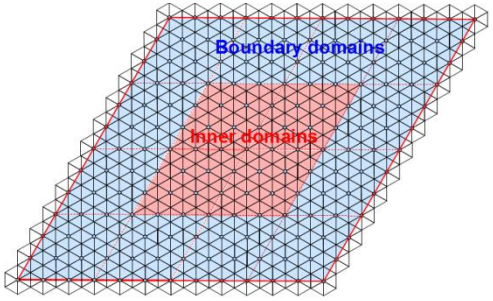
$$\frac{\partial u}{\partial t} + \frac{\overline{\frac{\partial u}{\partial \eta}}^v \overline{W}^{vh}}{\overline{m}^h} + \overline{(f + \nabla_\eta \times u) \times u}^{TRiSK} + \nabla_\eta \left(\overline{\frac{u^2}{2}}^h + \Phi \right) + \overline{\theta}^h \nabla_\eta \pi = S_u$$

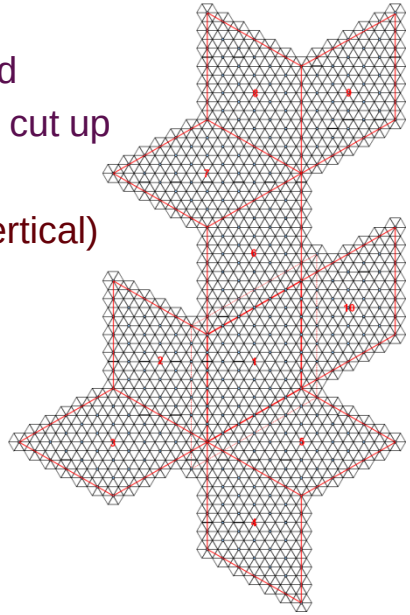


- Solve "primitive" equations on the sphere
- New numerical conservative schemes on staggered grid
  - TRISK scheme (Thuburn et al., 2010)

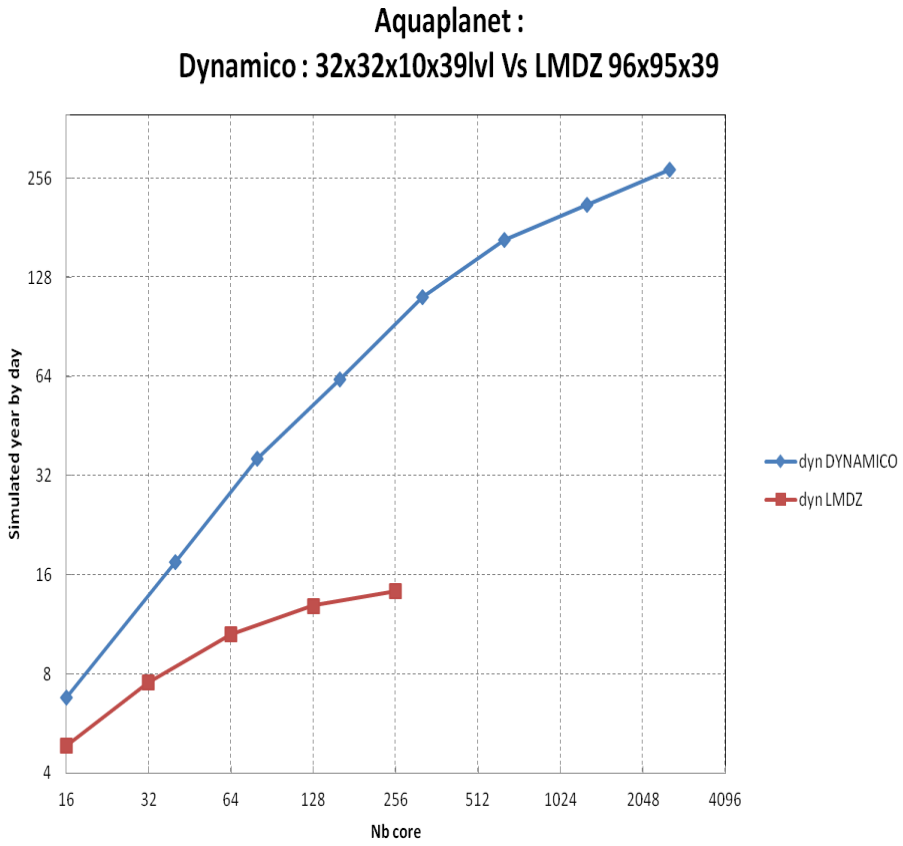
Parallelism & HPC

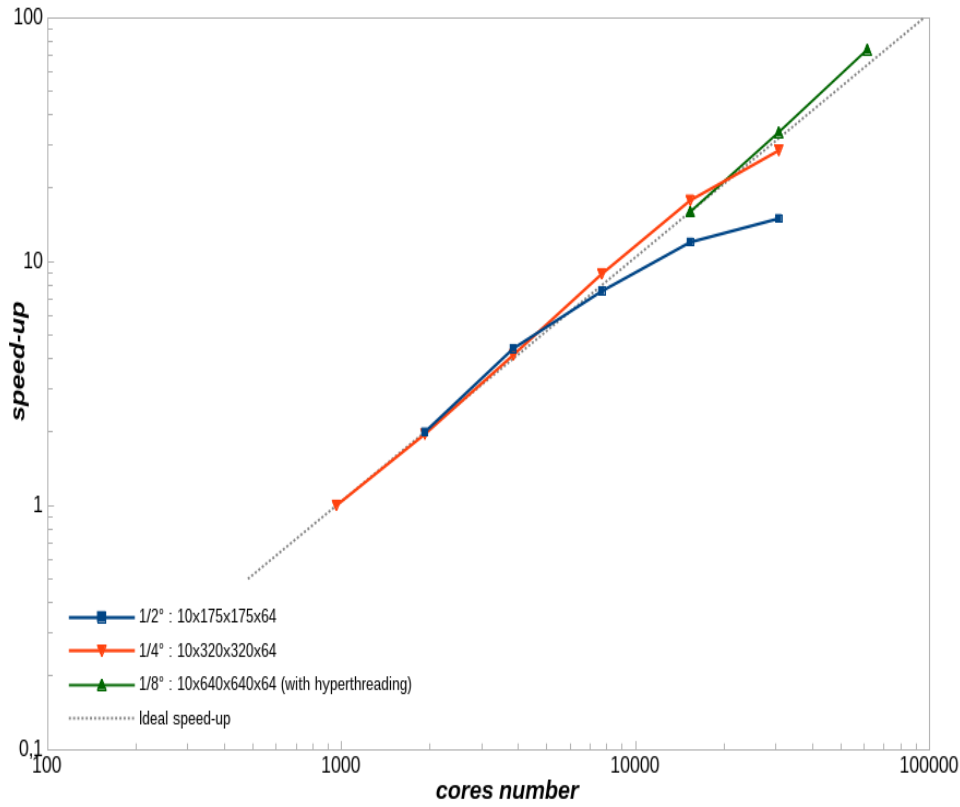
- Implementation on semi-structured grid
  - 10 regular rhombus tiles, hexagons paved
  - Regular memory access, domain easy to cut up
- 1 MPI parallelism level (horizontal)
- 2 OpenMP parallelism level (horizontal & vertical)
- 99.7 vectorized (~60% peak perf. On SX8)





DYNAMICO : an Highly SCALABLE MODEL





- Comparison with actual model (LMDZ)

Decreasing time restitution

- Factor 10 for low resolution
- Up to x40 for High resolution

CINES Big Challenges 2014

- Simulating high resolution Saturn atmosphere
  - 12 Millions hours on OXYGEN supercomputer
  - Resolution up to 1/8° degrees
  - High Scalability tested up to 60 000 cores (hyperthreading)

Conclusion
Impact on codes

Conclusion

- Les progrès en calcul haute performance boostent les domaines classiques des simulations numériques et en ouvrent de nouveaux
- Pour tirer parti de ces potentialités, il faut revoir en profondeur nos méthodes numériques : le parallélisme inter and intra nœud massif, l'utilisation efficace des unités vectorielles, la localité, sont nécessaires et le resteront dans un futur prévisible
- Les méthodes à forte intensité calculatoire tirent un meilleur parti de la puissance de calcul et évitent le goulet de la bande passante
- L'adaptation des méthodes numériques a commencé, mais il reste un travail important pour tirer parti de la nouvelle génération de processeurs manycore
- Certains algorithmes, comme le Monte Carlo, les méthodes particulaires, les calculs sur grilles régulières sont naturellement adaptées aux machines actuelles et futures
- Le Monte Carlo a l'avantage supplémentaire de répondre au problème de résilience (pannes et erreurs hardware)
- Les calculs d'incertitude basés sur des méthodes de Monte Carlo ont un bel avenir…

Commissariat à l'énergie atomique et aux énergies alternatives
Centre DAM Ile-de-France | 91297 Arpajon Cedex, France
T. +33 (0)1 69 26 40 00

DAM
DIF
ED

Etablissement public à caractère industriel et commercial | RCS Paris B 775 685 019

The big trends towards the Exascale

- Hardware
  - Reduce the electrical consumption of the compute units
    - Ever more cores with a reasonable frequency
    - Systematic usage of specialized units SIMD/SIMT
  - Reduce the cost of data movements
    - Unavoidable placement of CPUs and powerful SIMD units inside the same chip
    - Integration of high performance network interfaces to the chip (SoC)
    - Stacked memory to increase bandwidth
    - More threads per cores to hide latencies
- Software
  - Necessary evolution of standards to cope with hardware evolution
    - Don't rush on new fancy options, yet learn by experimenting on prototypes
  - Upgrade of the full software ecosystem
    - Compilers, debuggers, profilers
- Challenges
  - Programming those processors at more than 10% of their peak
  - Optimize existing codes to fully utilize the platforms

Multithread your code

Vectorize your algorithms

Use all units

Work on data location(s)

Multithread your code

MPI as usual (PGAS maybe one day?)