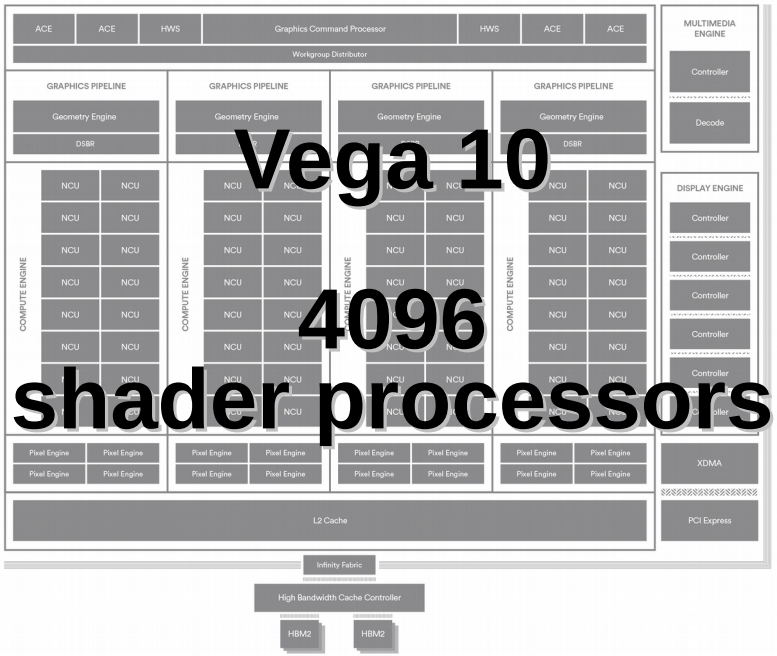
CBP
CENTRE BLAISE PASCAL

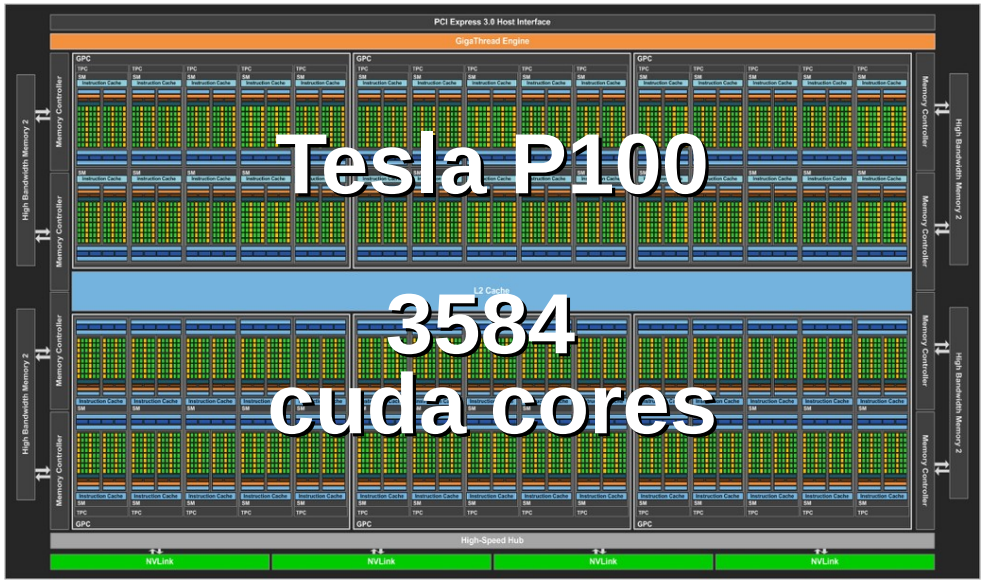
Le GPU à l'épreuve… de la métrologie

"Les illusions tombent l'une après l'autre, comme les écorces d'un fruit. Et le fruit, c'est l'expérience. Sa saveur est amère ; elle a pourtant quelque chose d'âcre qui fortifie." Gérard de Nerval, Les filles de feu

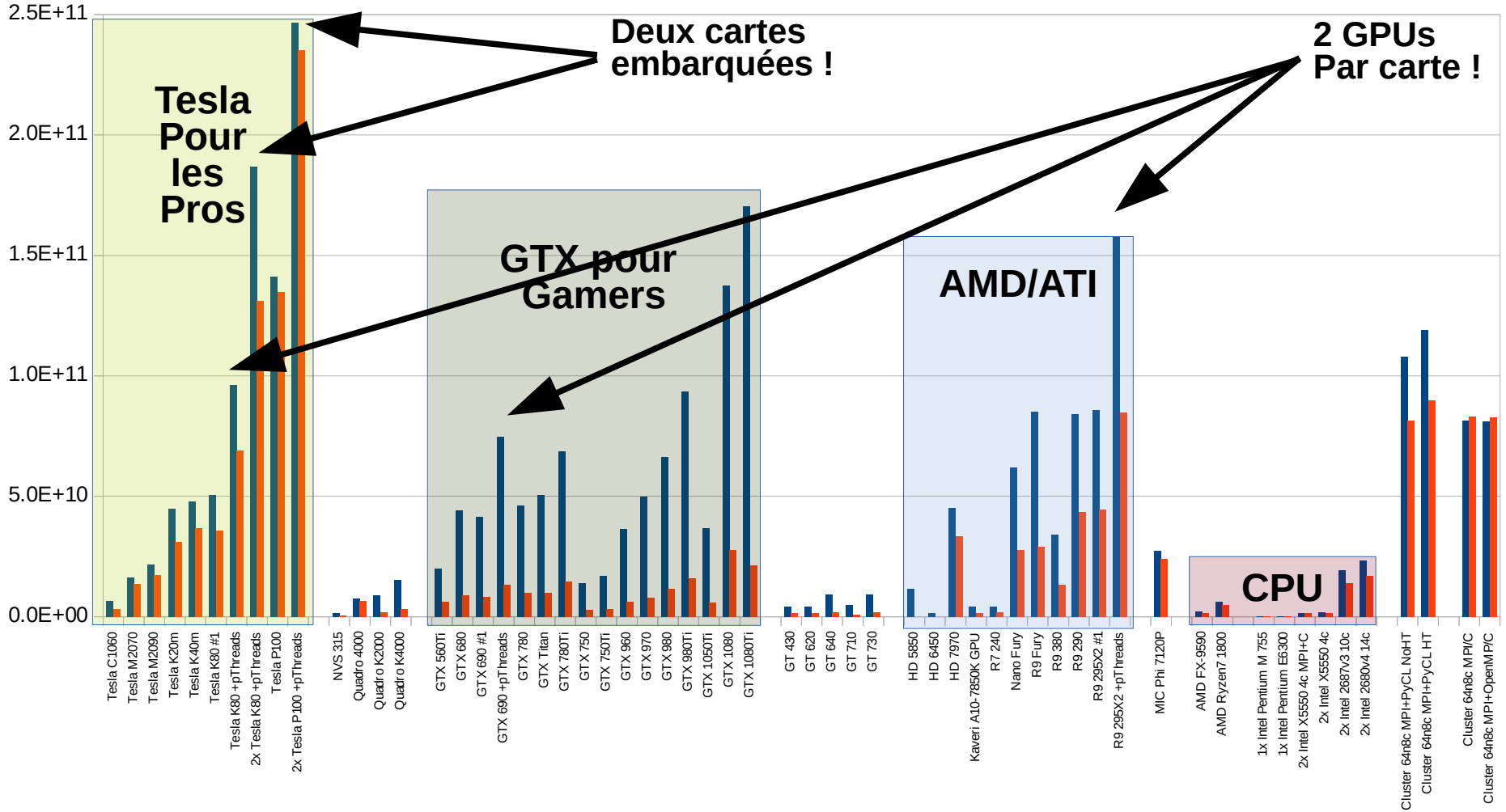
Emmanuel Quémener

Petits rappels : la différence GPU/CPU



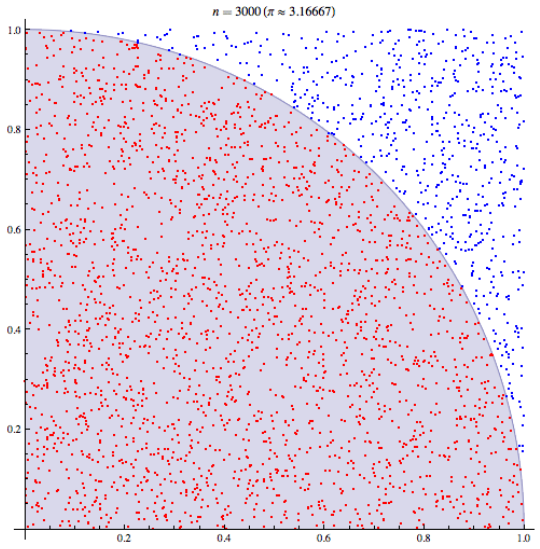


Le match Pi Monte Carlo
Python vs C & CPU vs GPU vs Phi



Un code « test » qui se parallélise « bien » : Le « Pi Dart Dash » ou « Pi Monte Carlo »

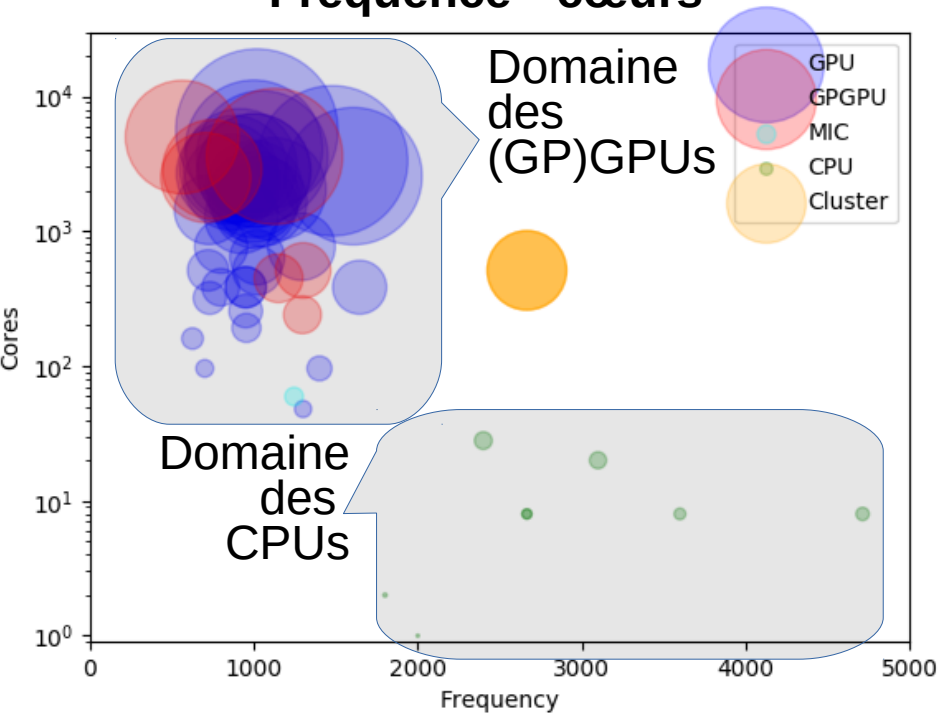
- Exemple classique du calcul de Monte Carlo
- Implémentation parallèle : distribution
  - De 2 à 4 paramètres
    - Nombre total d'itérations
    - Régime de Parallélisme (PR)
    - (Type de variable : INT32, INT64, FP32, FP64)
    - (RNG : MWC, CONG, SHR3, KISS)
  - 2 observables simples :
    - Estimation de Pi estimation (juste indicative, Pi n'est pas rationnel :-))
    - Temps écoulé



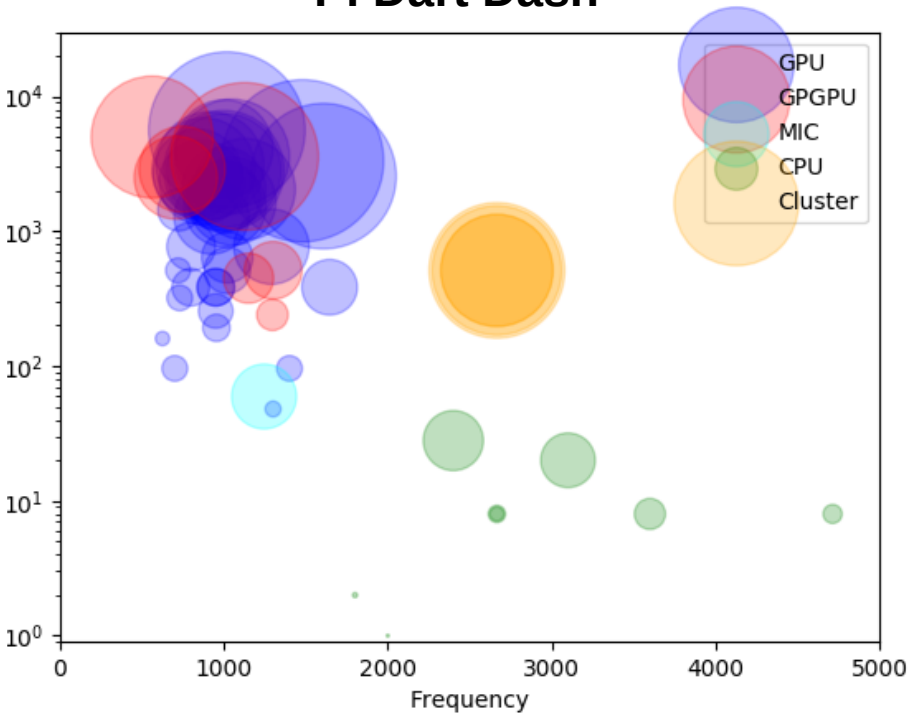
- Représentatif d'un code scientifique ?

Représenter les performances... Question de battement, de cœurs ?

Performance Théorique
Fréquence * cœurs



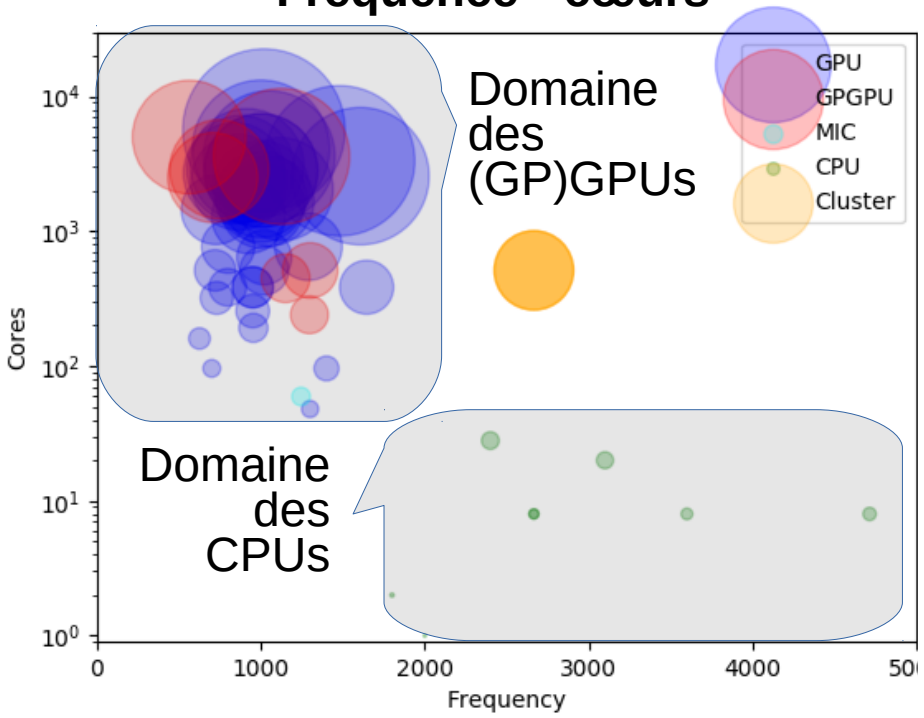
Performance en 32 bits
“Pi Dart Dash”



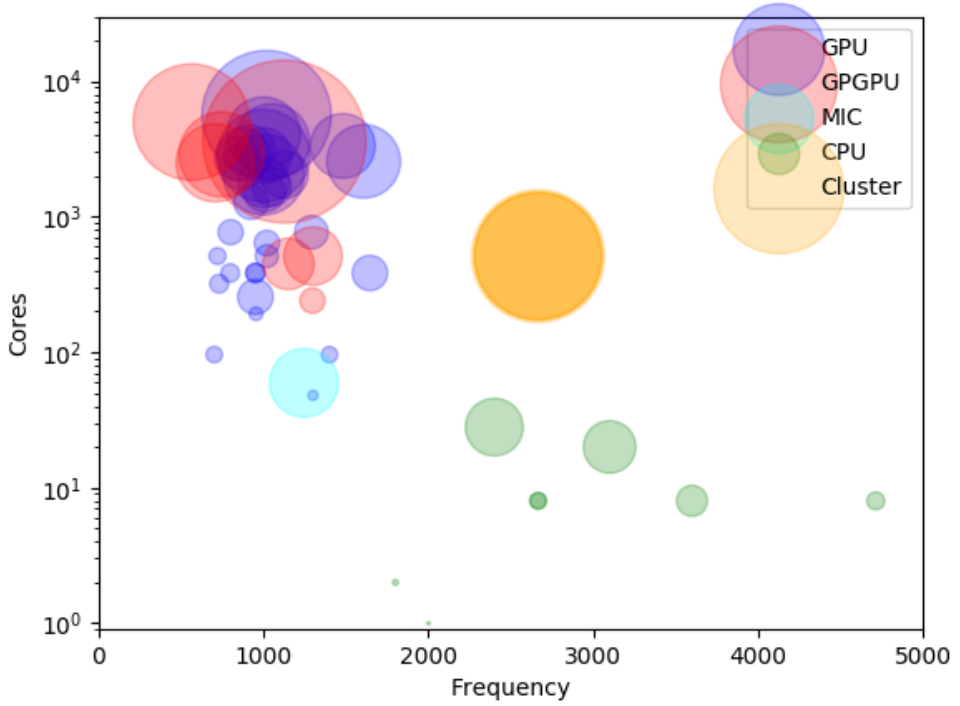
- Sur un GPU, cohérence entre performances théorique & pratique
- Sur un CPU, performance relative meilleure

La performance en informatique
Question de battement, de cœurs, et de bits !

Performance Théorique
Fréquence * cœurs

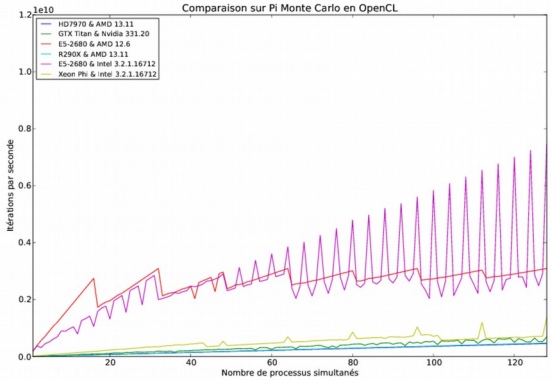


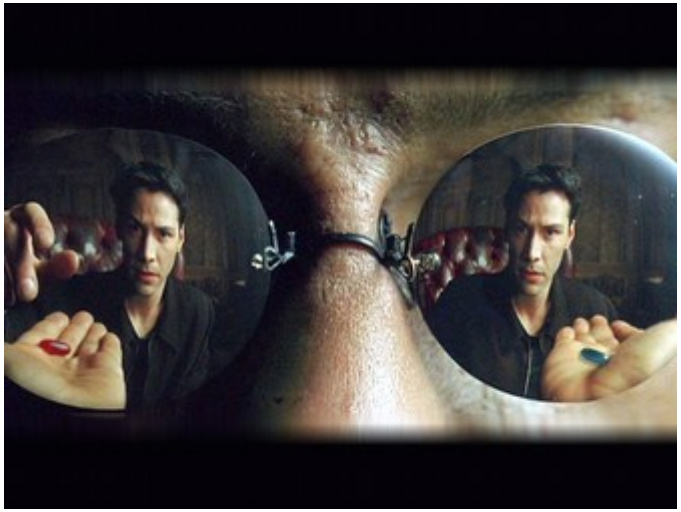
Performance en 64 bits
“Pi Dart Dash”

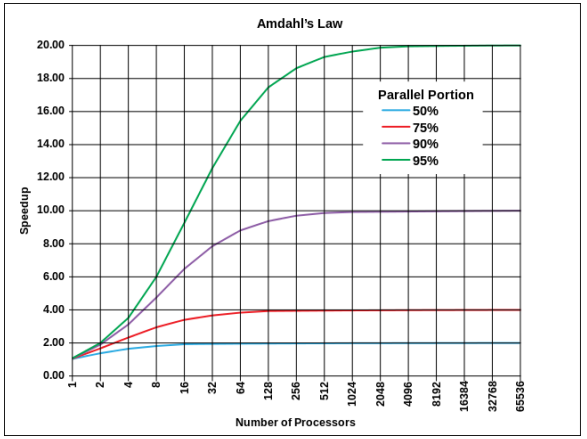


- Sur un GPU, baisse très sensible des performances pour les GPU
- Sur un CPU, performance relative encore meilleure

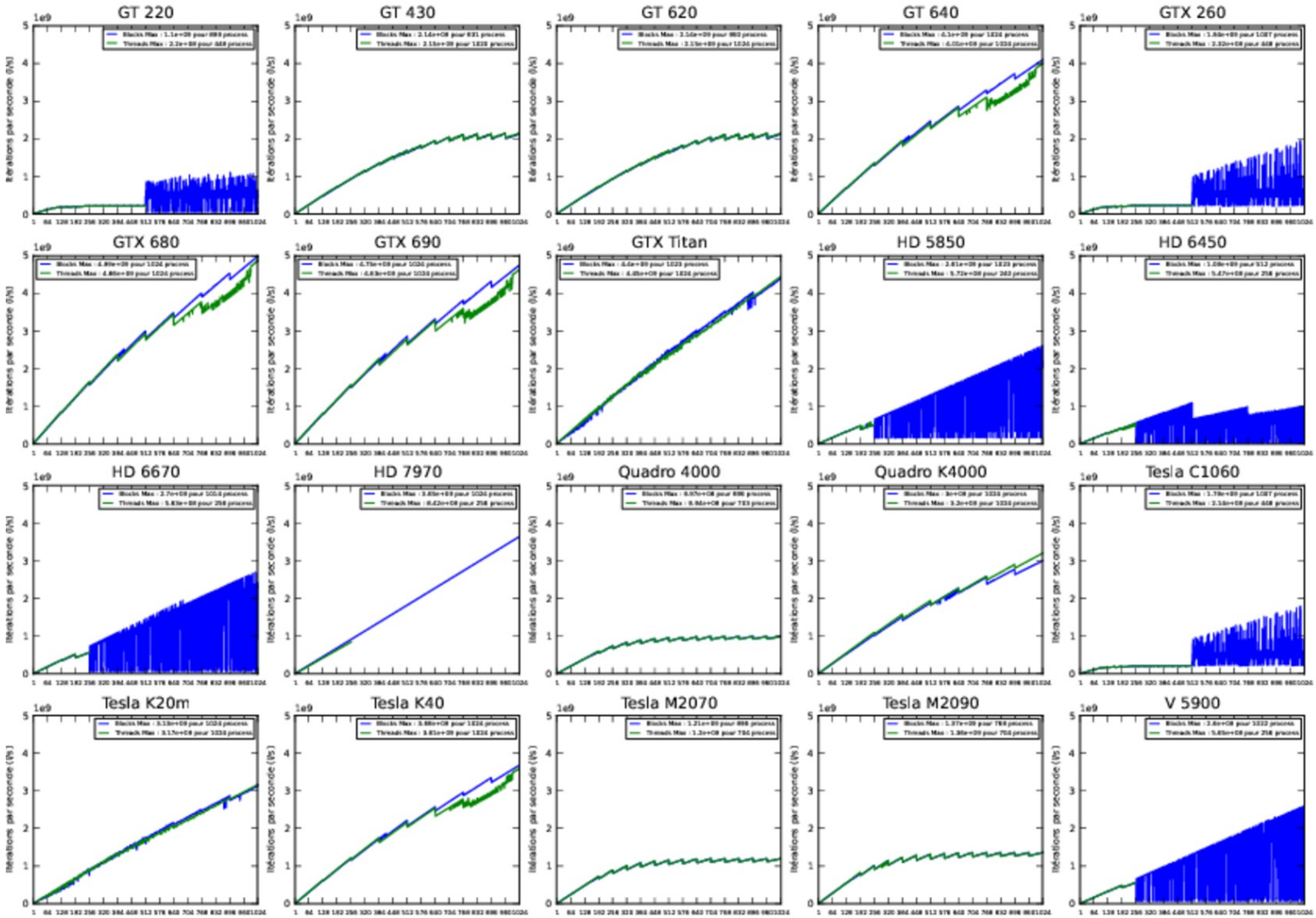
Prêt à prendre la « pilule rouge » ?





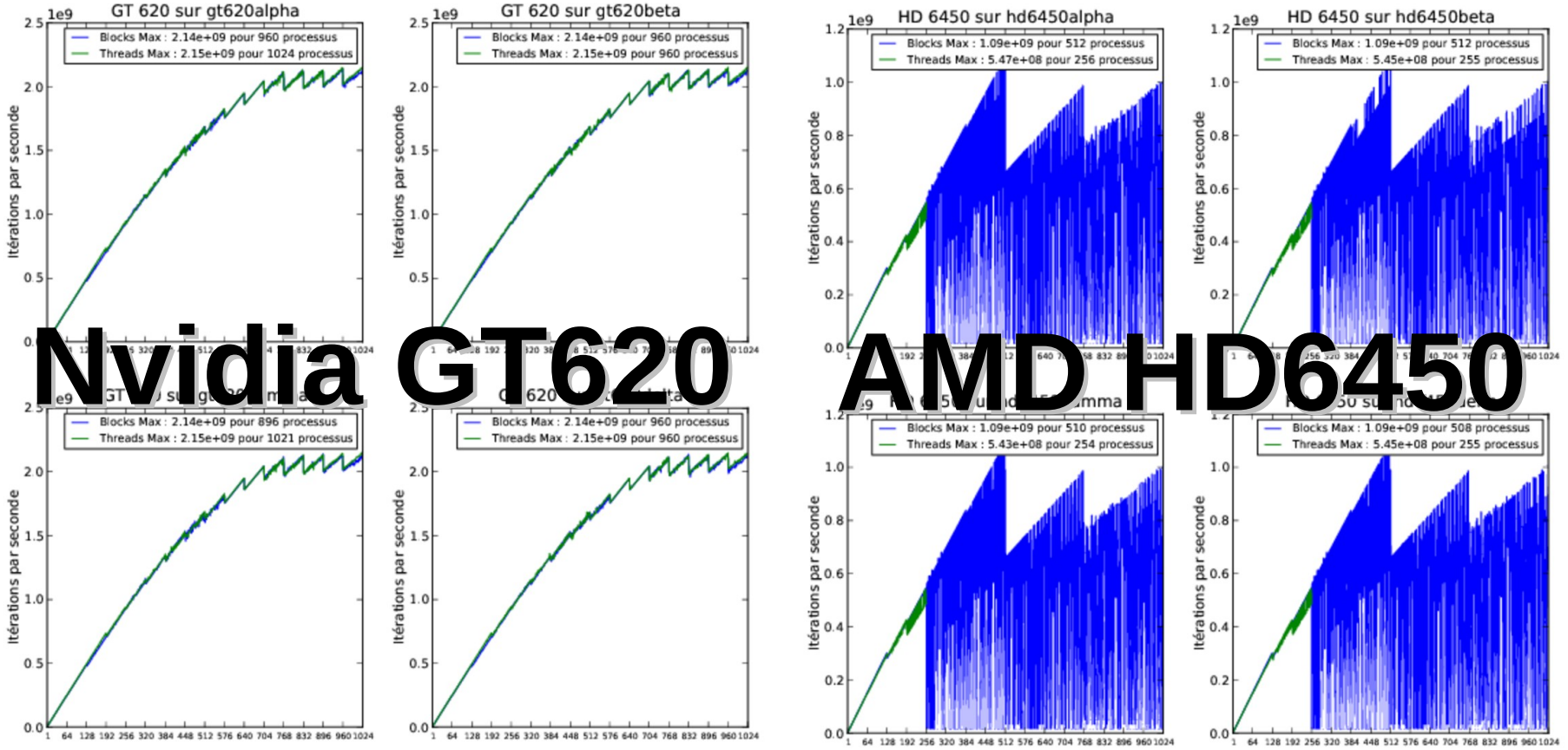


Comportement de 20 (GP)GPU vs PR



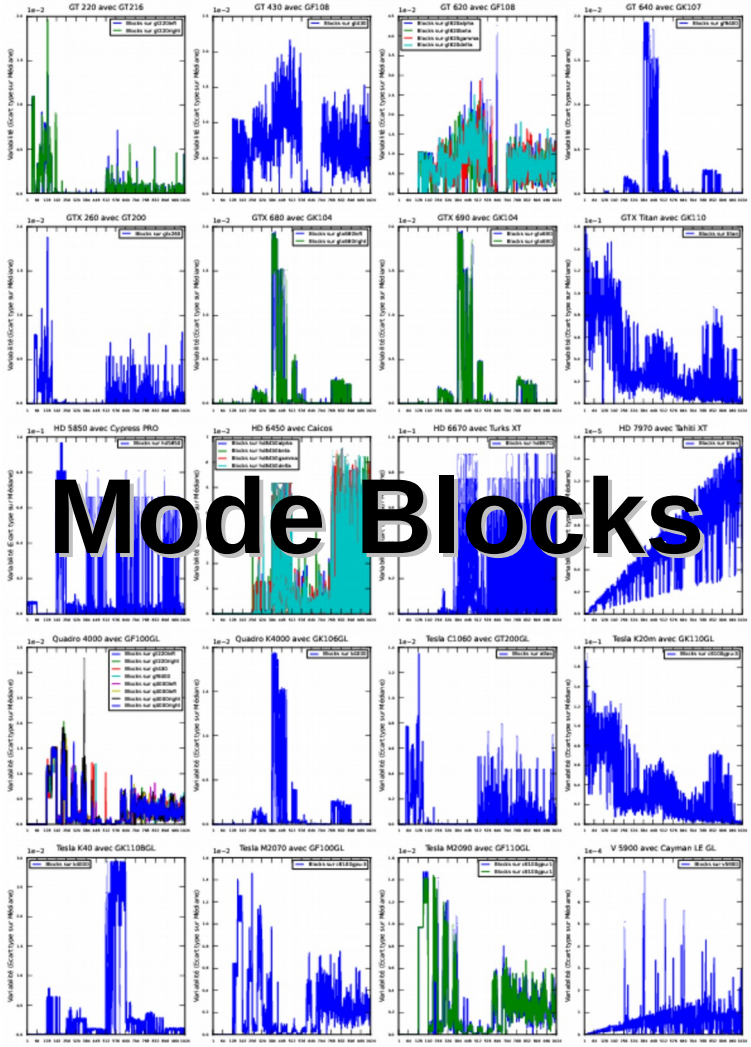
Emmanuel QUÉMENER CC BY-NC-SA

Enveloppes de parallélisme reproductibles ? Pour deux cartes spécifiques...

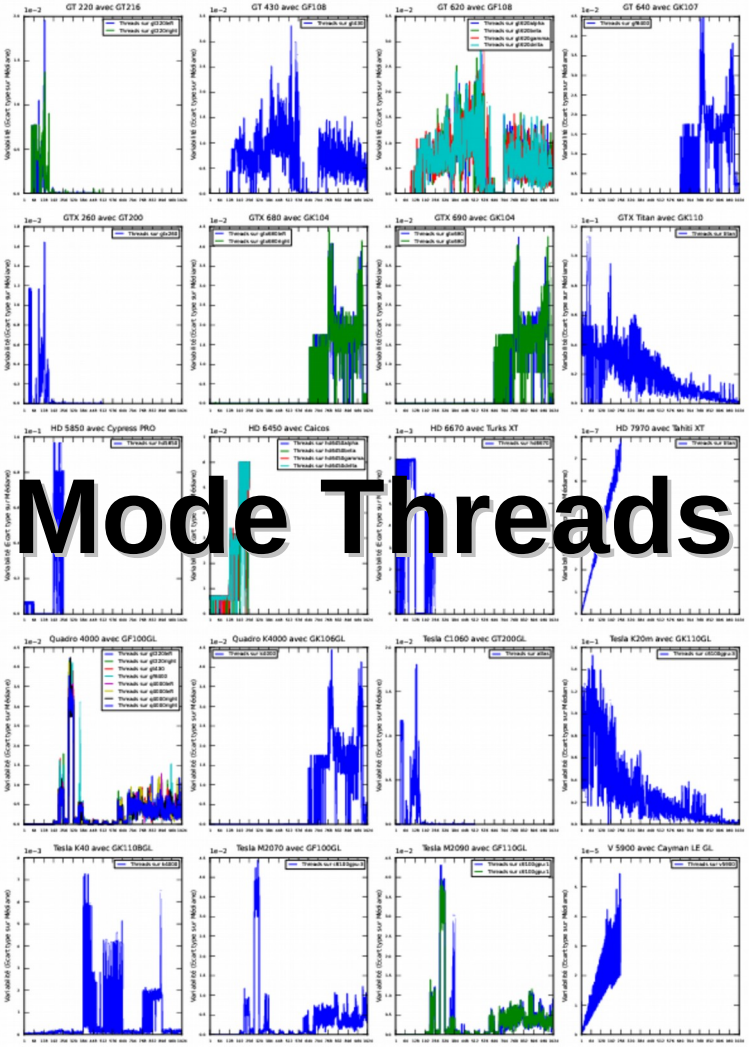


Nvidia GT620 **AMD HD6450**

Variabilité : un critère distinctif ! Quelle étrange propriété :-/ ...

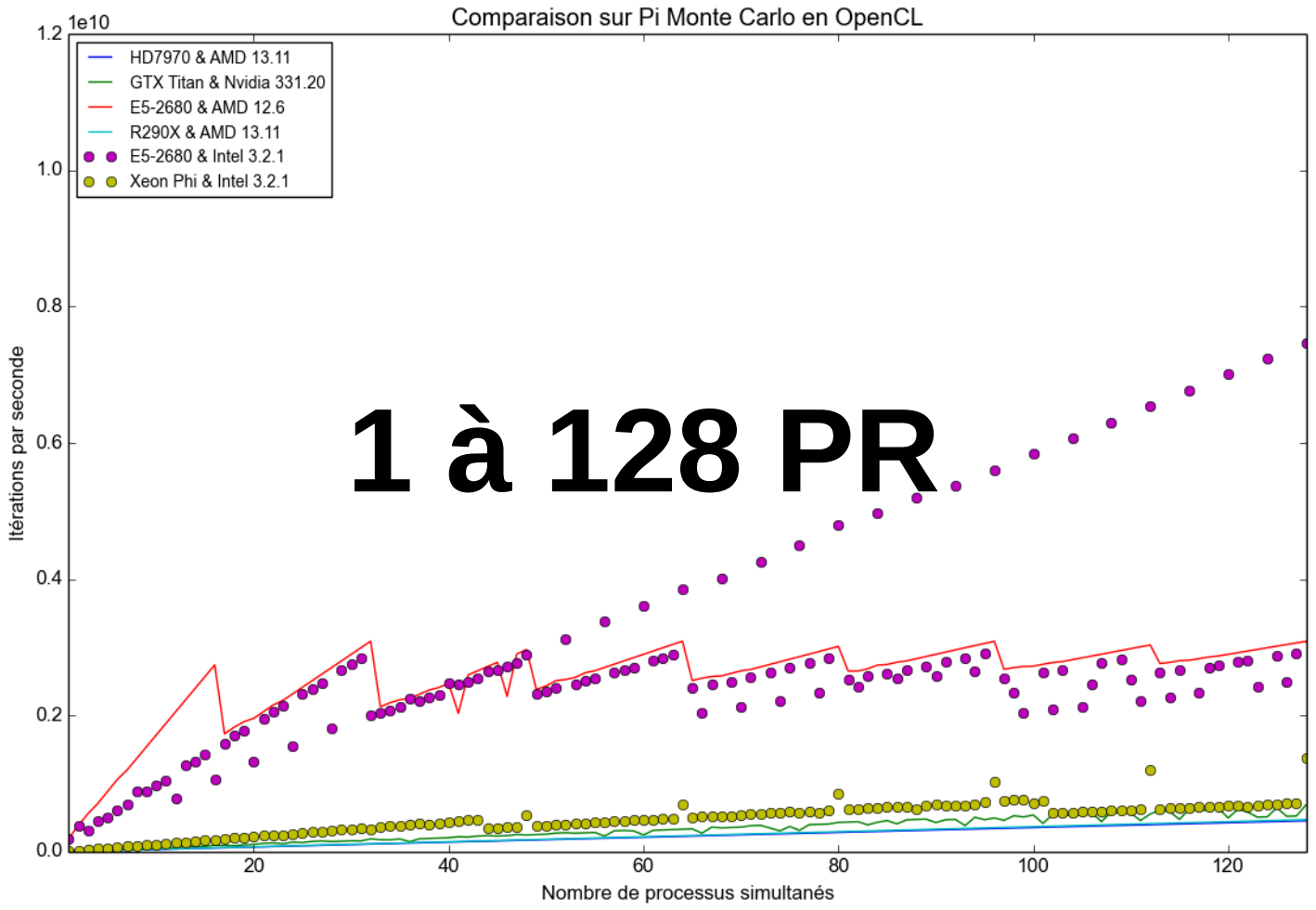


Mode Blocks

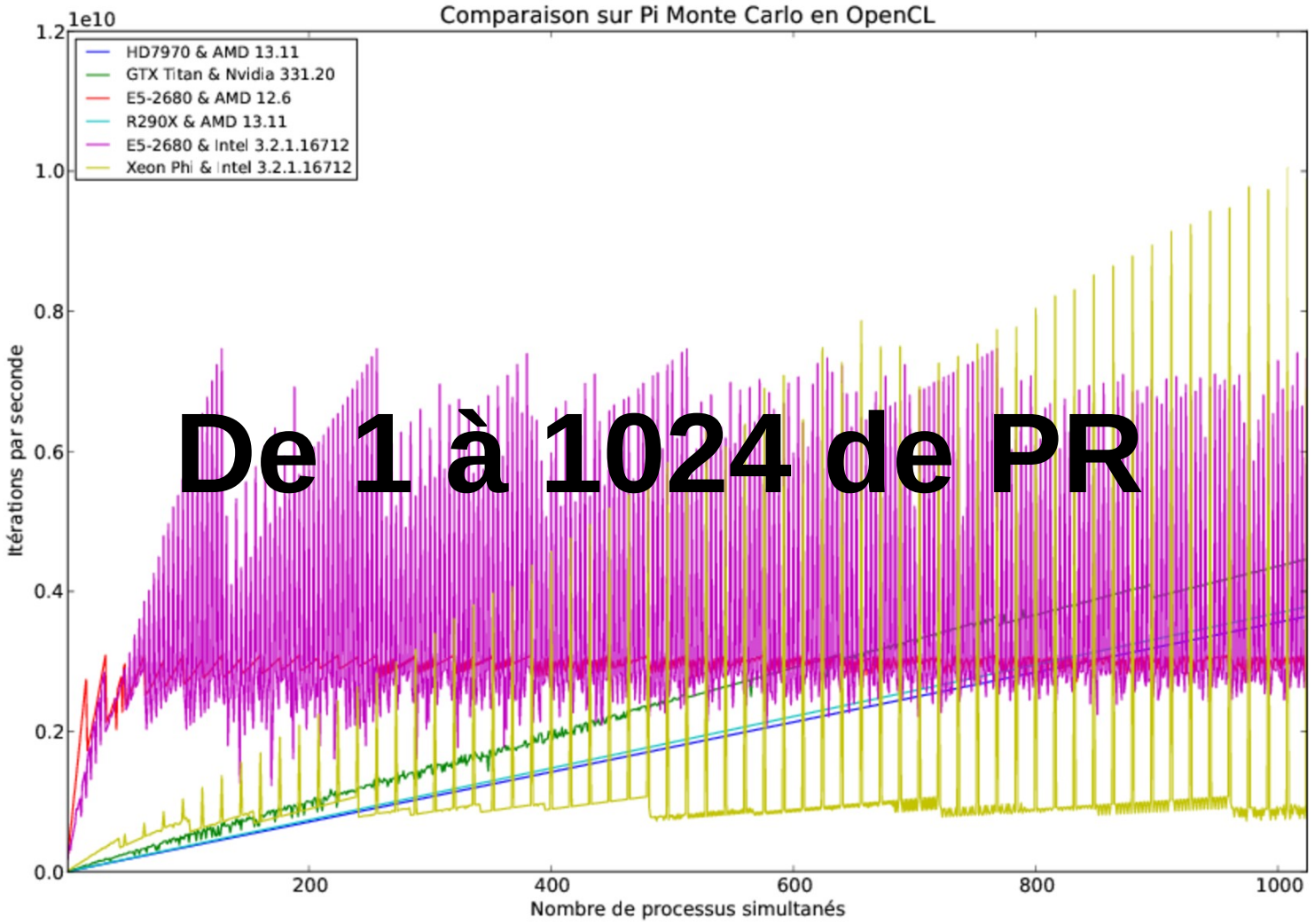


Mode Threads

Petite comparaison entre *PU
Bas pararégime : le règne du CPU

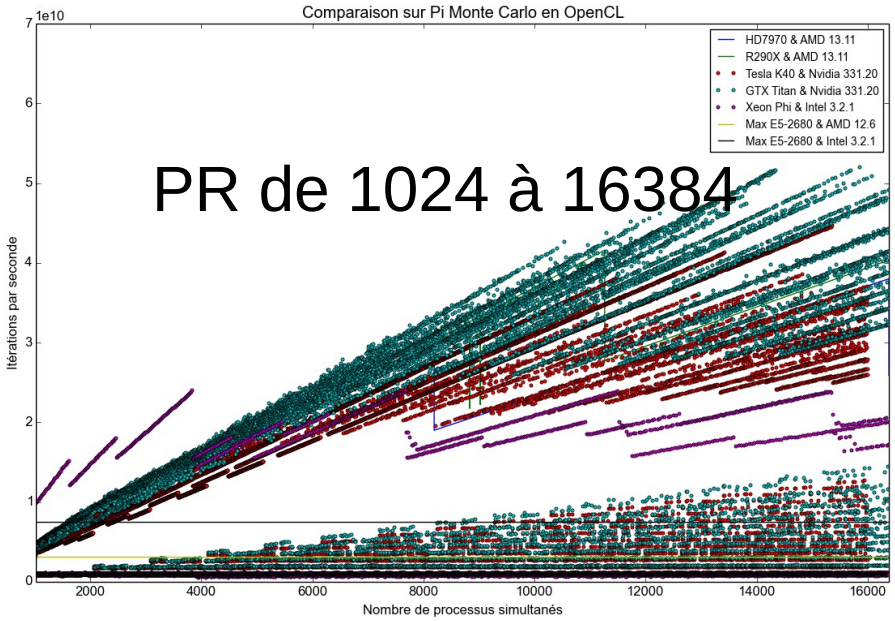


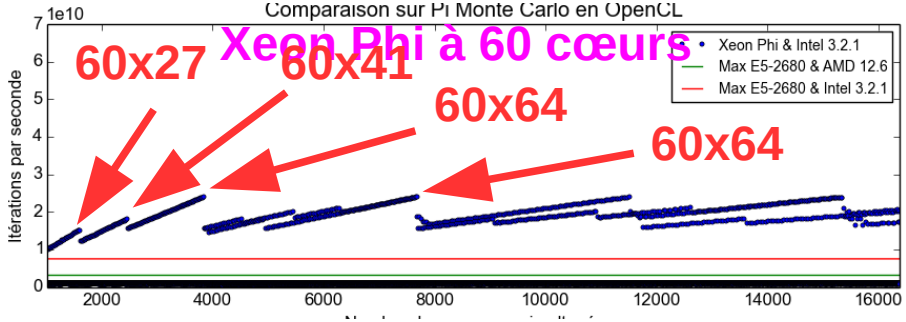
Les processeurs toujours efficaces
Le Xeon Phi « sort du bois »

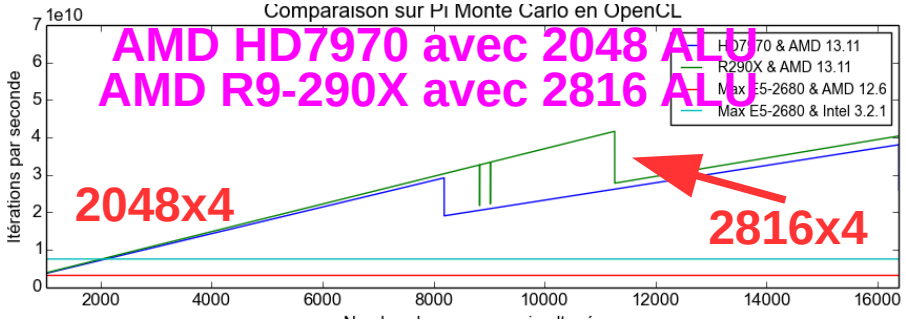


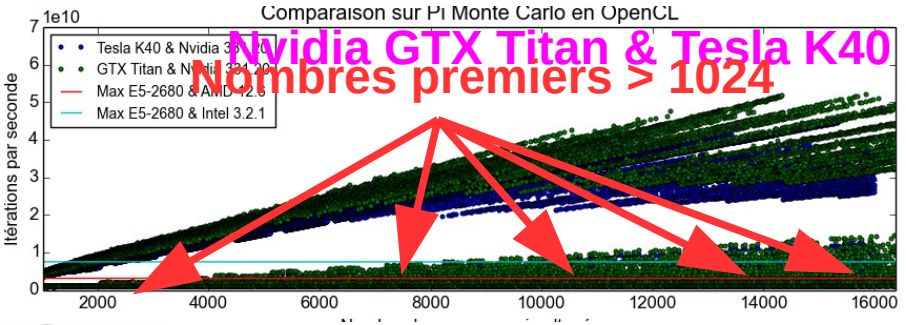
De 1 à 1024 de PR

Exploration profonde de PR ParaRégime de 1024 à 16384

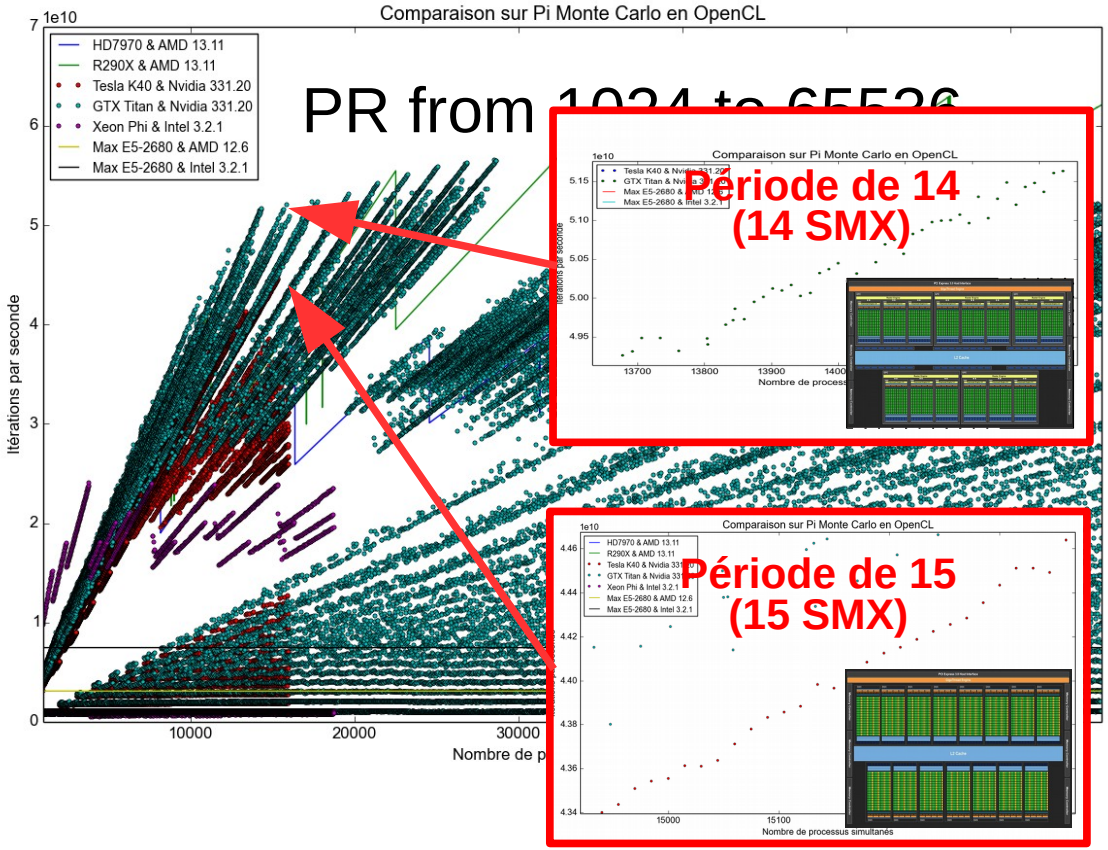




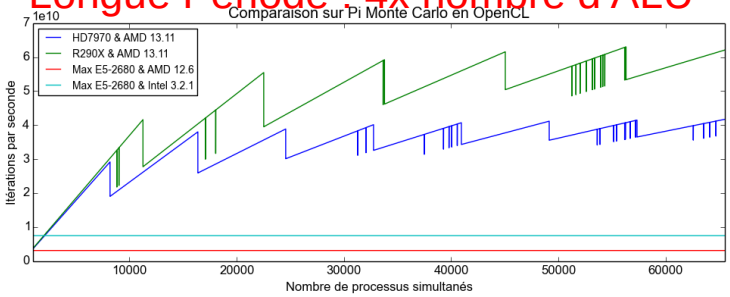




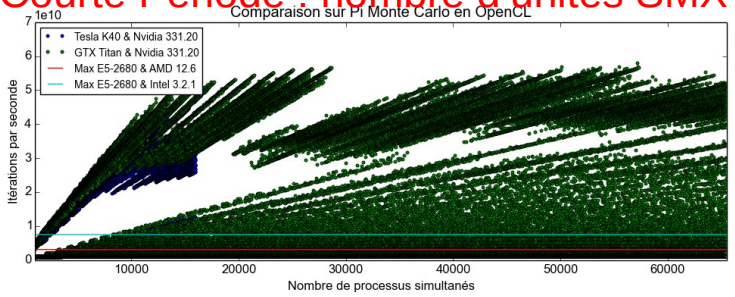
Investiguer la structure interne Par l'exécution de Pi Dart Dash



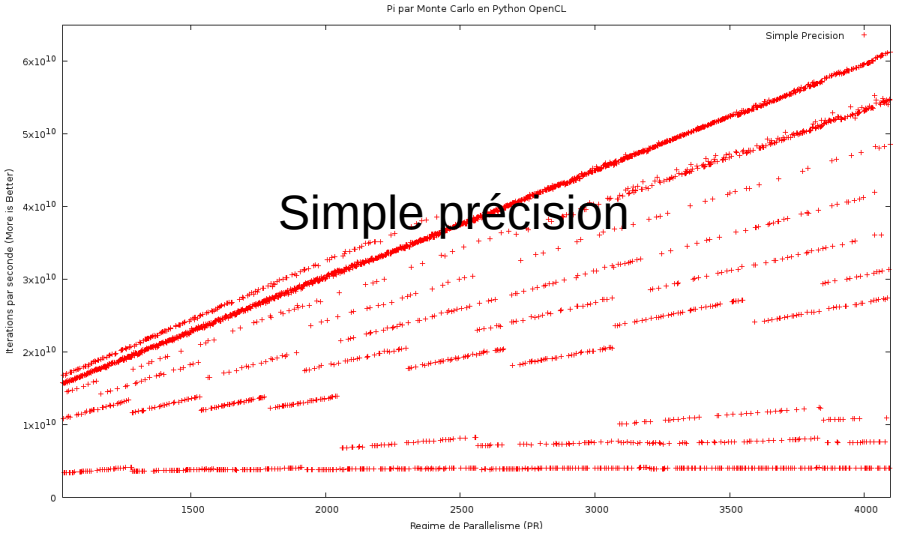
AMD HD7970 & R9-290
Longue Période : 4x nombre d'ALU



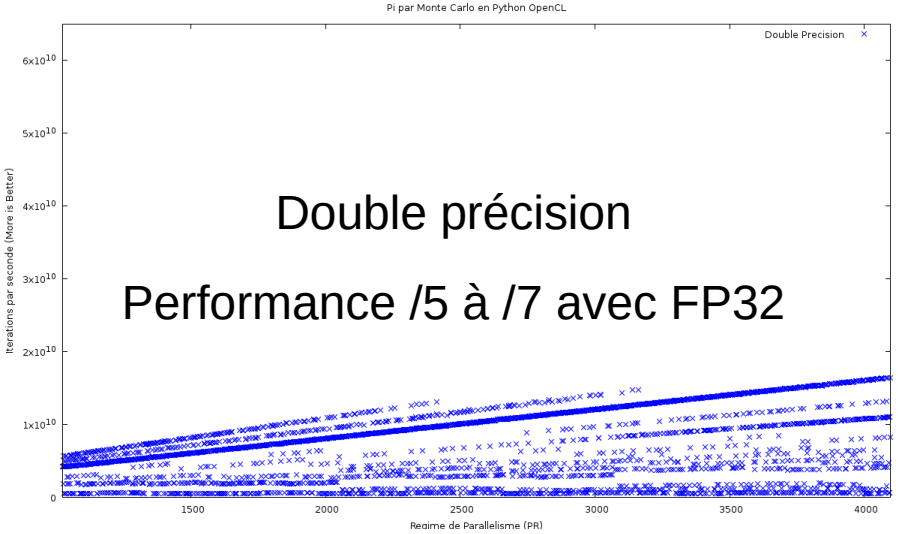
Nvidia GTX Titan & Tesla K40
Courte Période : nombre d'unités SMX



Et les dernières Nvidia GTX1080(Ti) Toujours affectées de bizarreries ?

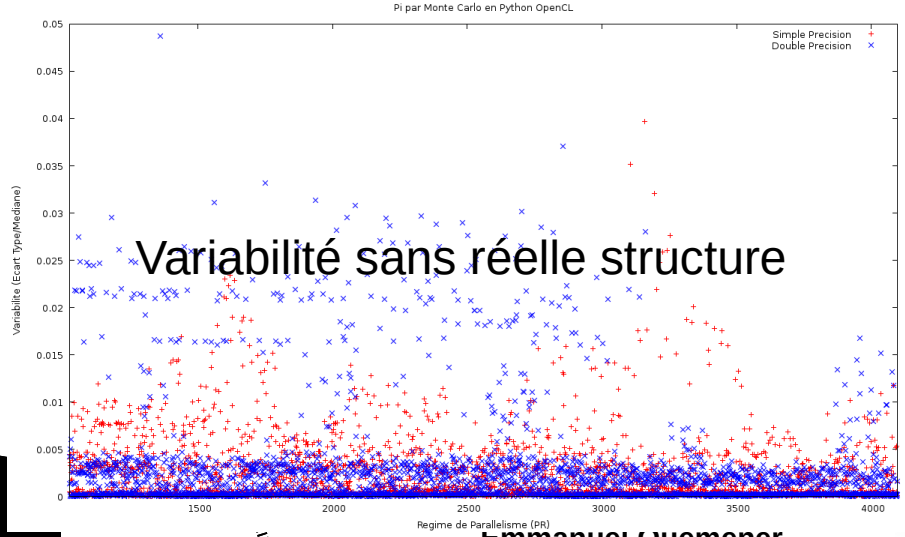


Simple précision

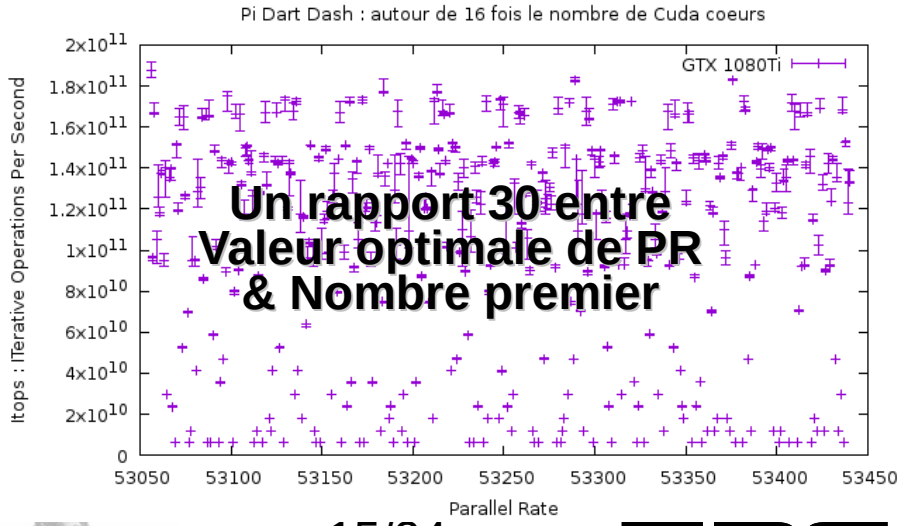


Double précision

Performance /5 à /7 avec FP32

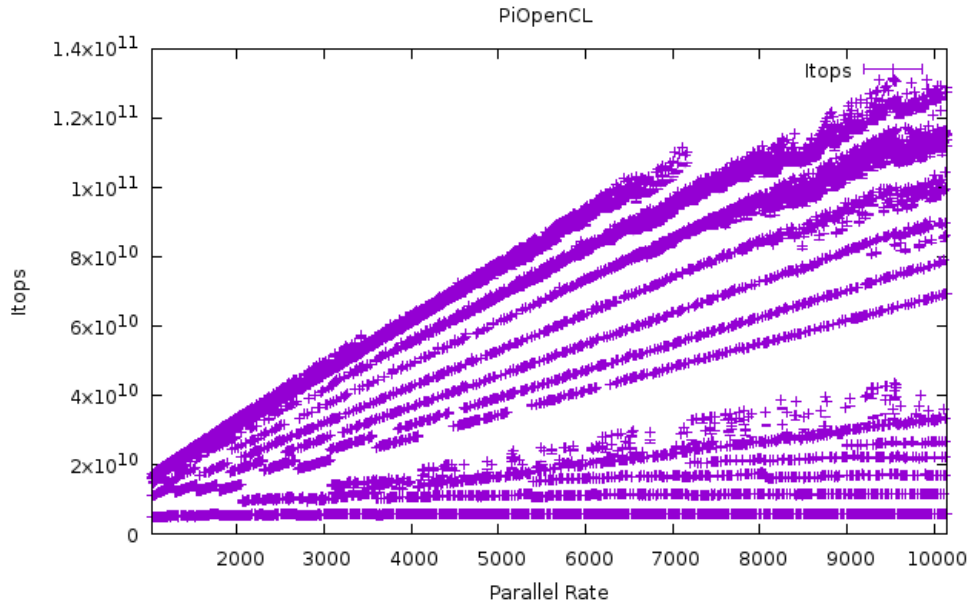


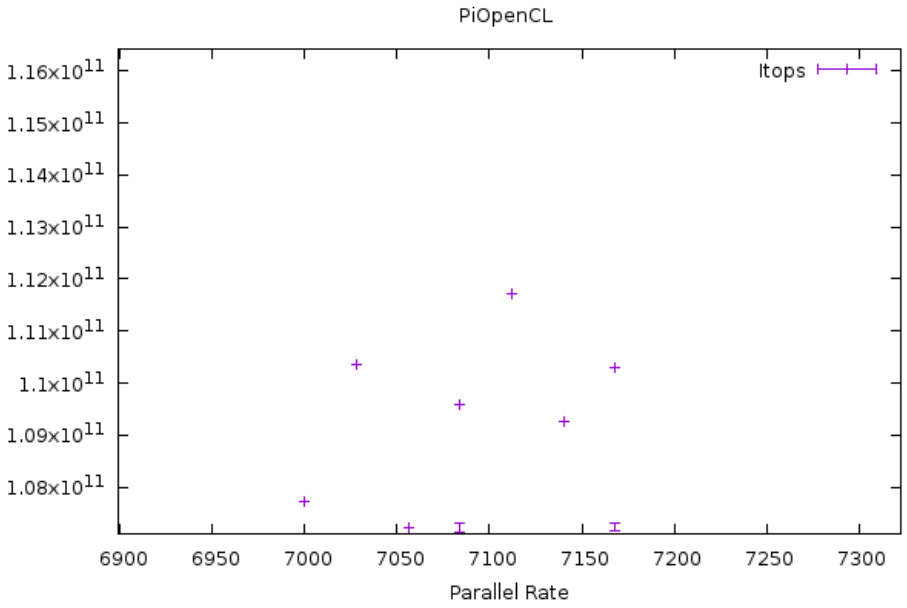
Variabilité sans réelle structure



Un rapport 30 entre Valeur optimale de PR & Nombre premier

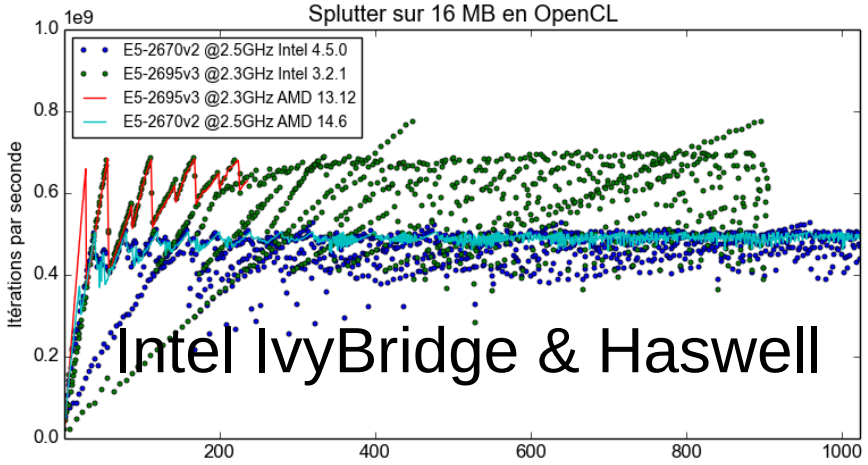
Et en passant sur un PiOpenCL en C pur ?

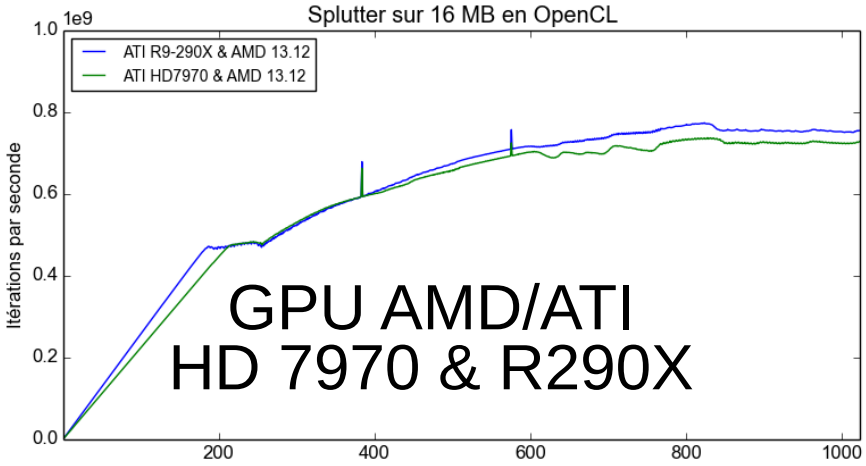


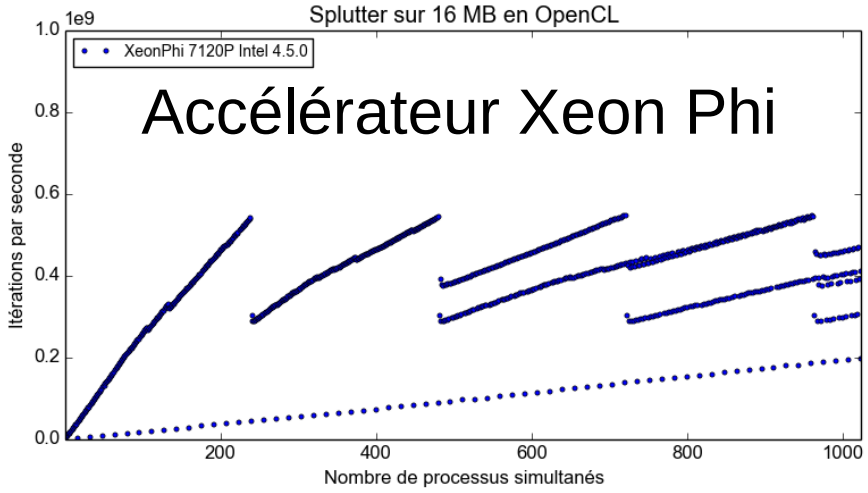


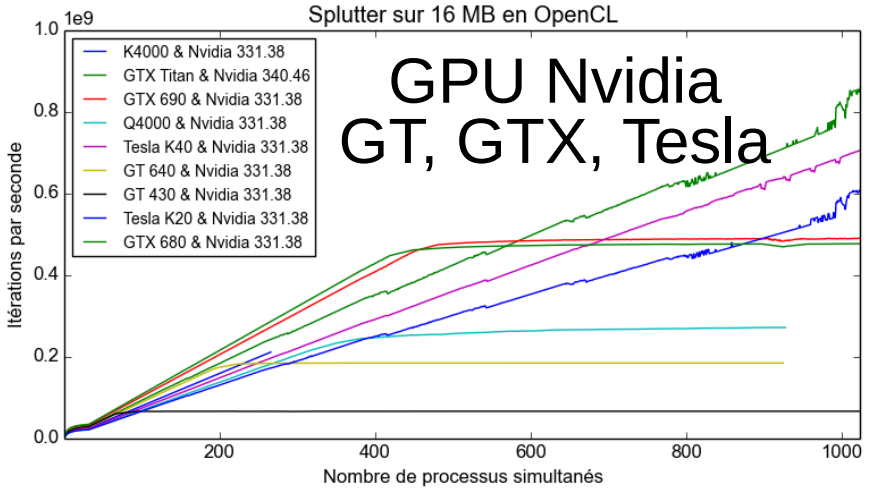
- Les mêmes détections de nombres premiers
- Les mêmes max (x2 le nombre de shaders)

Un test « memory bound »
Le « postillonneur » en OpenCL





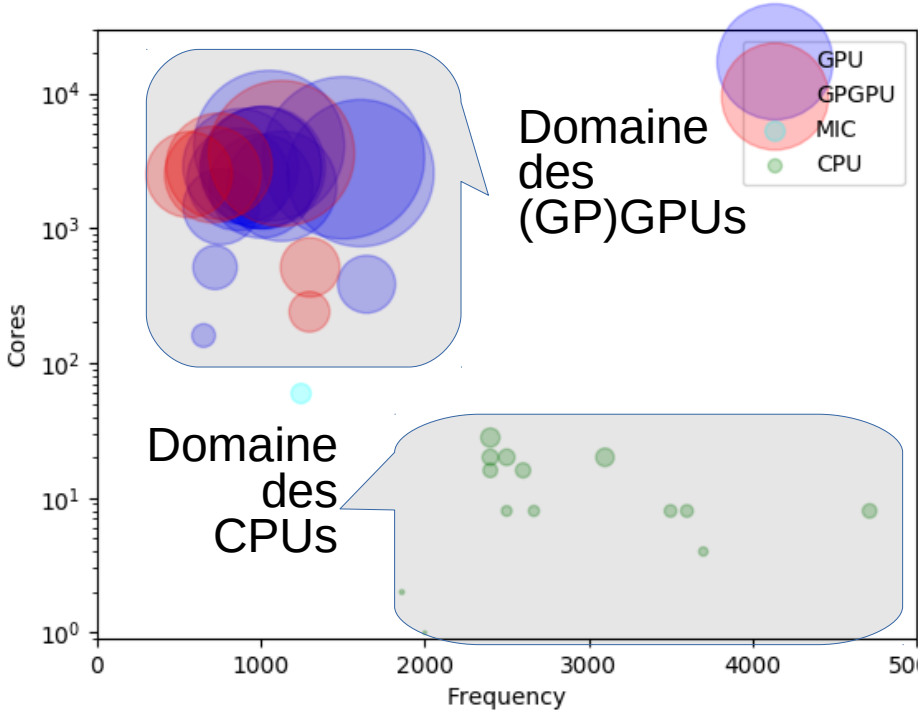


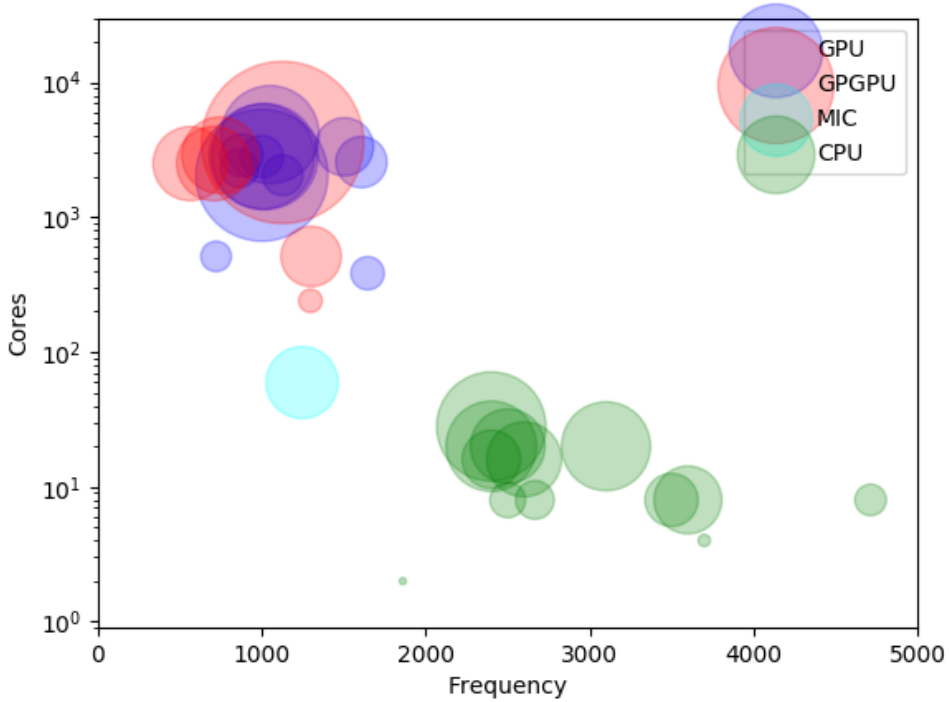


Retour à la physique
Un code newtonien N-corps...

- Passage sur un code « grain fin »
- Code N-body très simplement implémenté
  - Seconde loi de Newton, système autogravitant
- Méthodes d'intégration différentielle :
  - Euler Implicite, Euler Explicit, Heun, Runge Kutta
- Données d'entrées :
  - Physiques : nombre de particules
  - Numerique : pas d'intégration, nombre de pas
  - Architecture : influence de la nature du flottant FP32, FP64

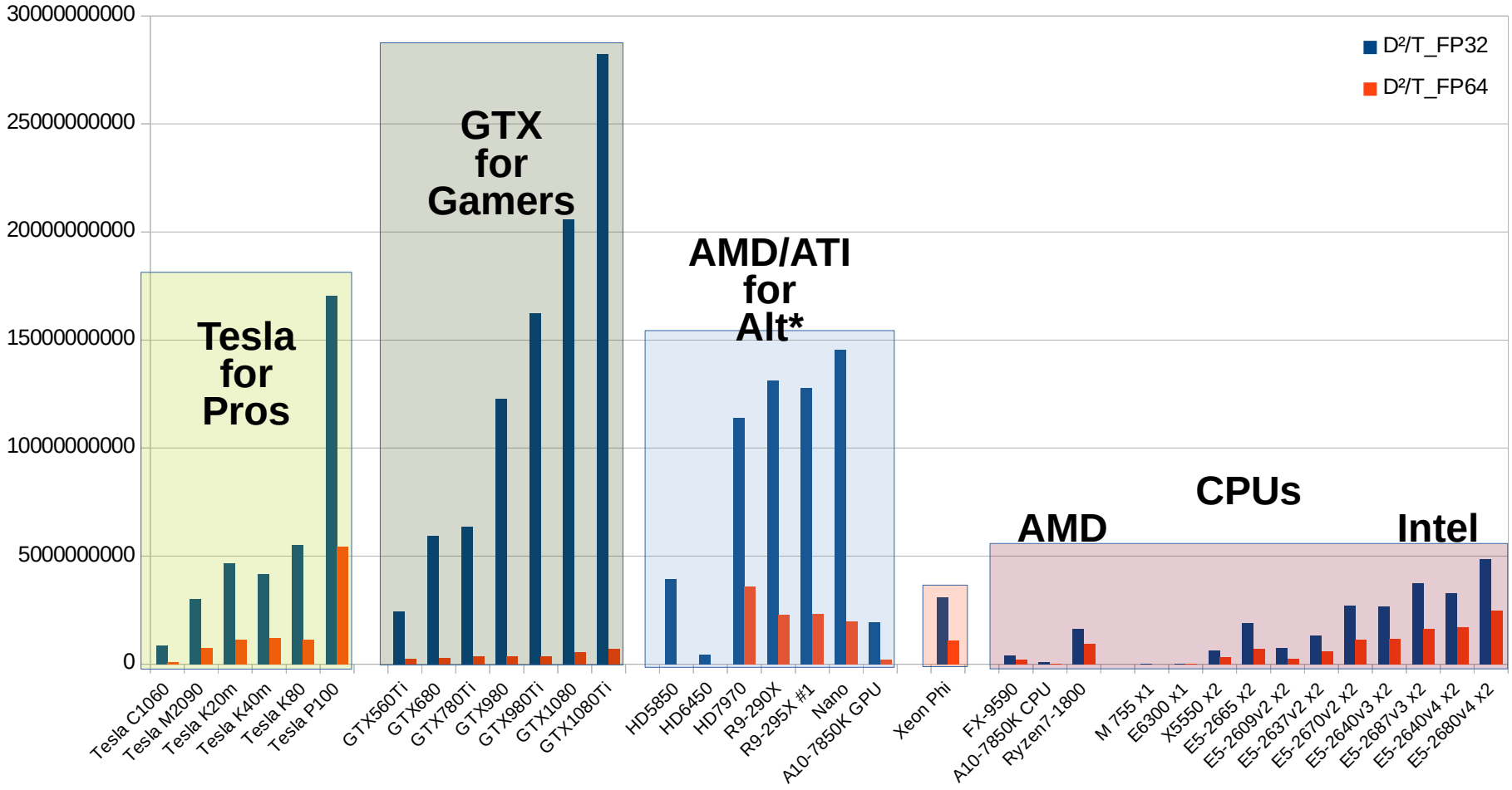
La performance en informatique
Pour un code plus physique...



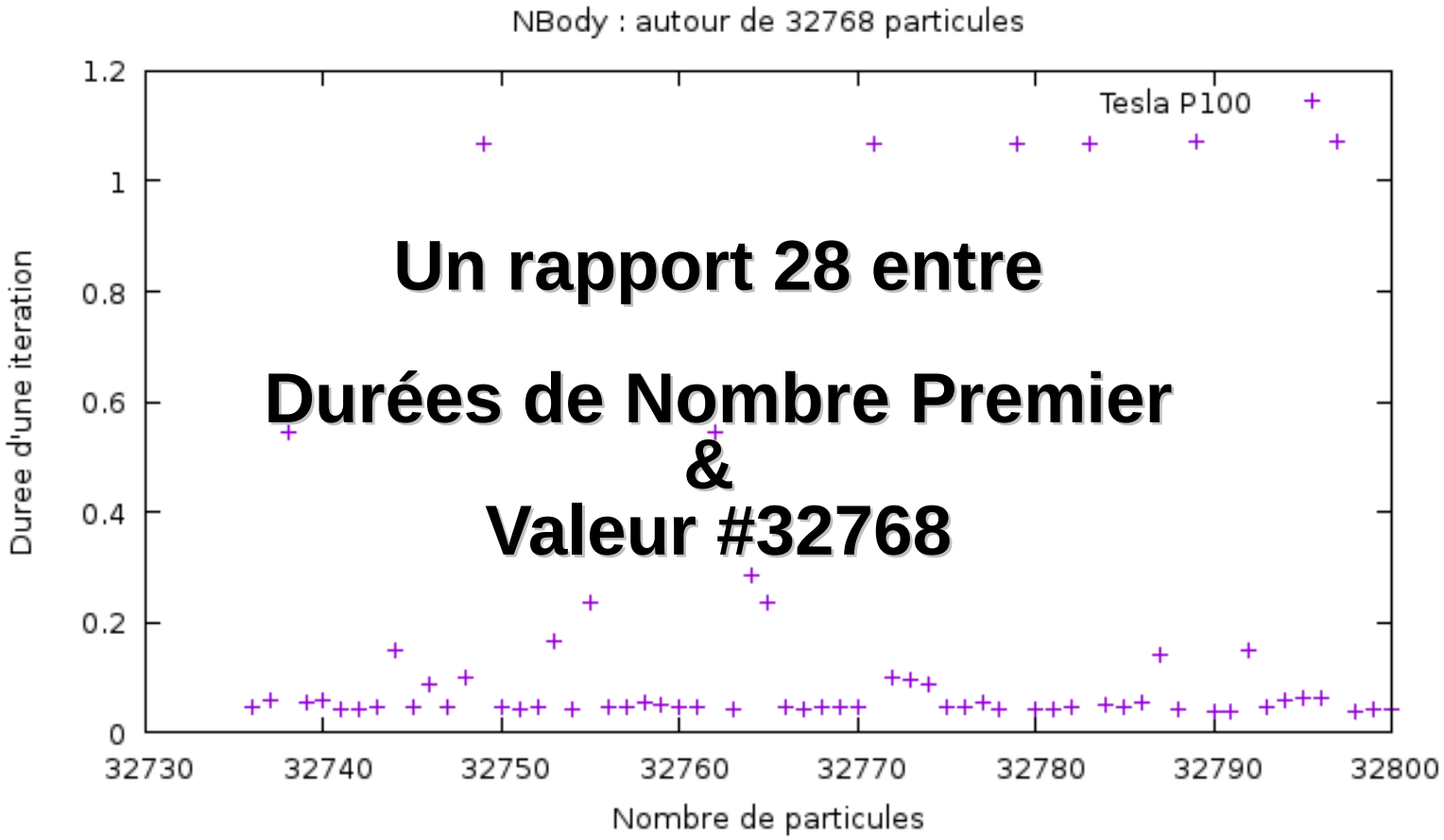


- Sur un GPGPU, performance stable
- Sur un GPU, baisse drastique de la performance (division par 20)
- Sur un CPU, performance relative meilleure

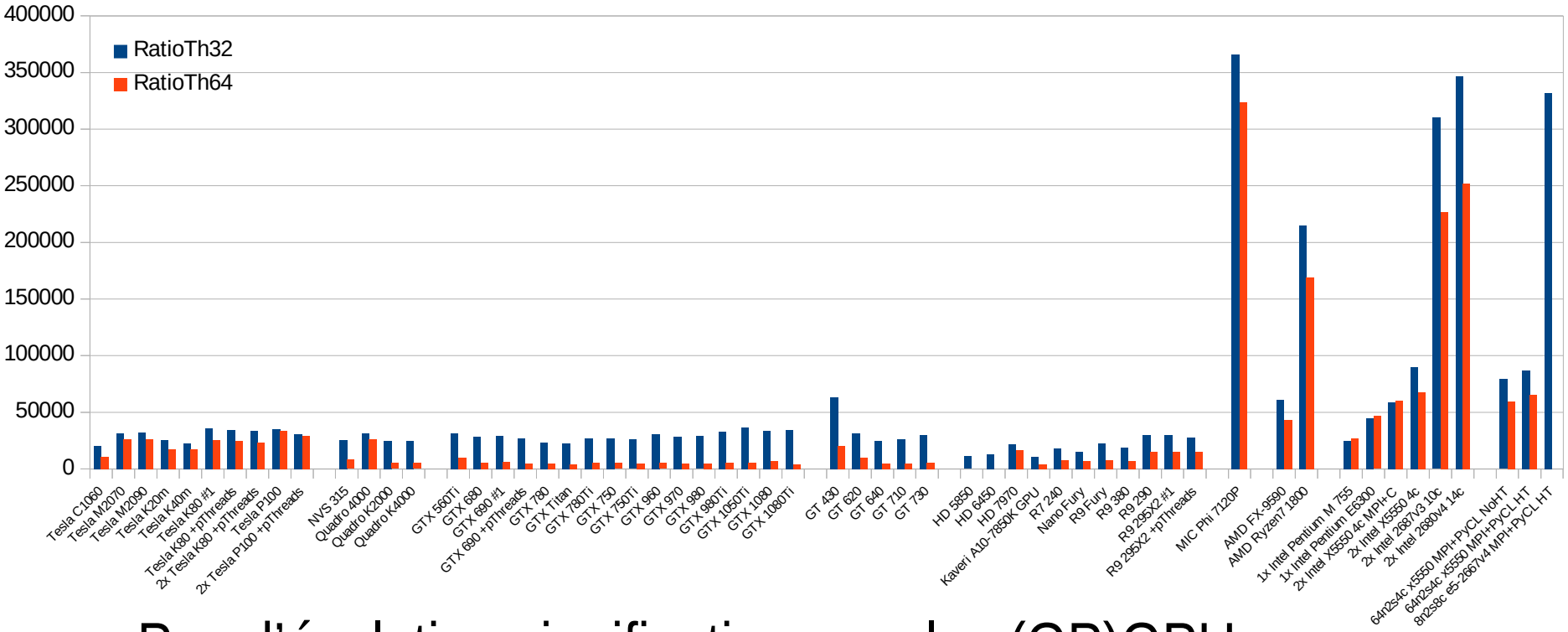
Modèle N-corps « naïf »
Une comparaison classique...



Et les effets « prime numbers » pour les cartes Nvidia ?

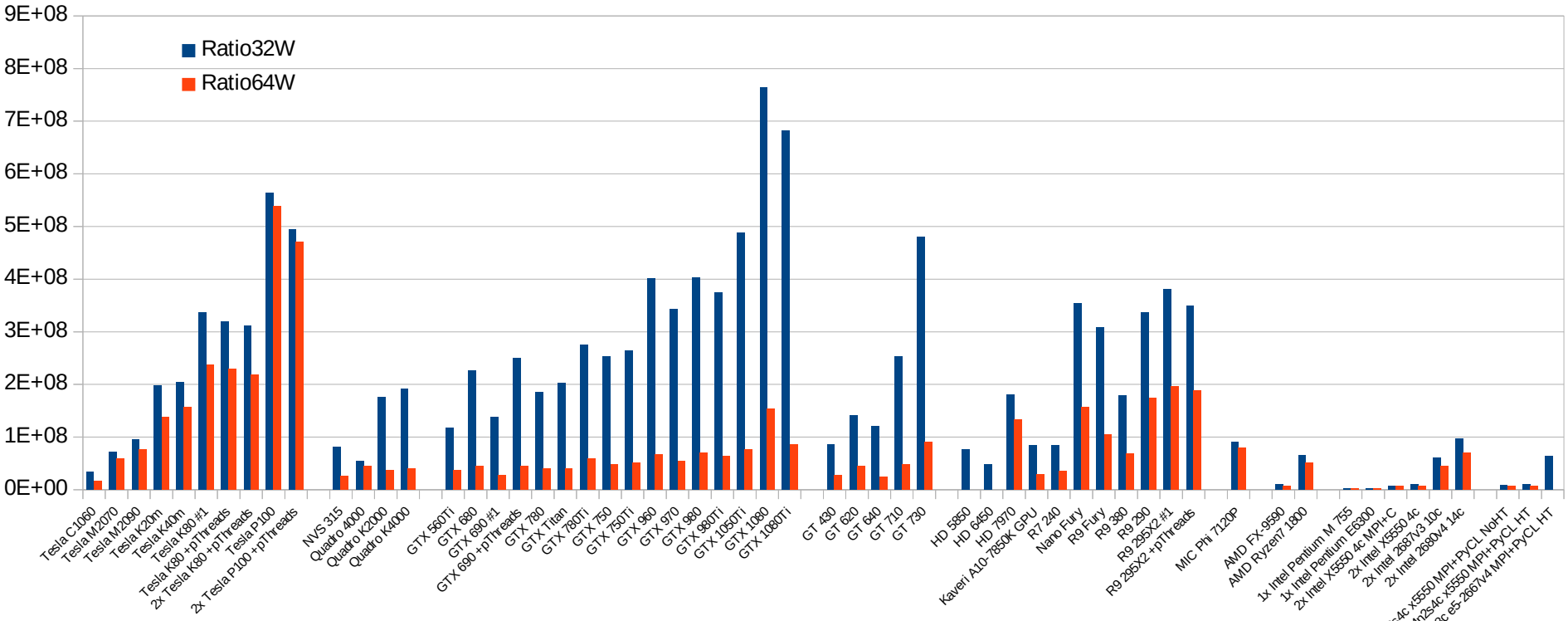


Performances normalisées : quel ratio ? Pi Dart Dash / Fréquence*Nb cœurs

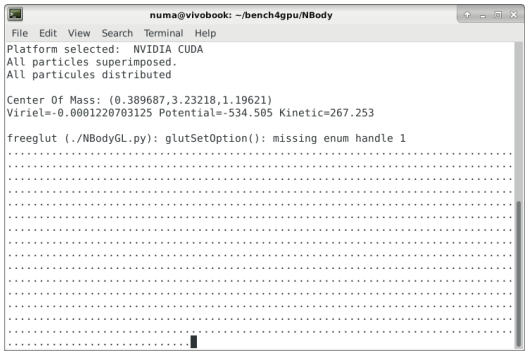
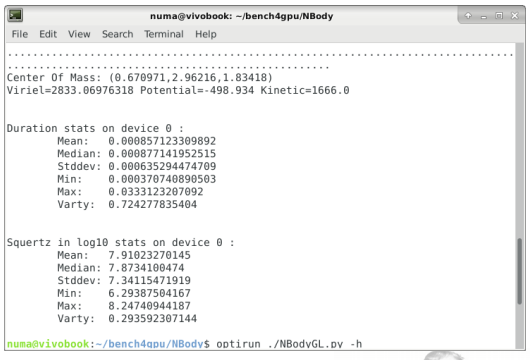
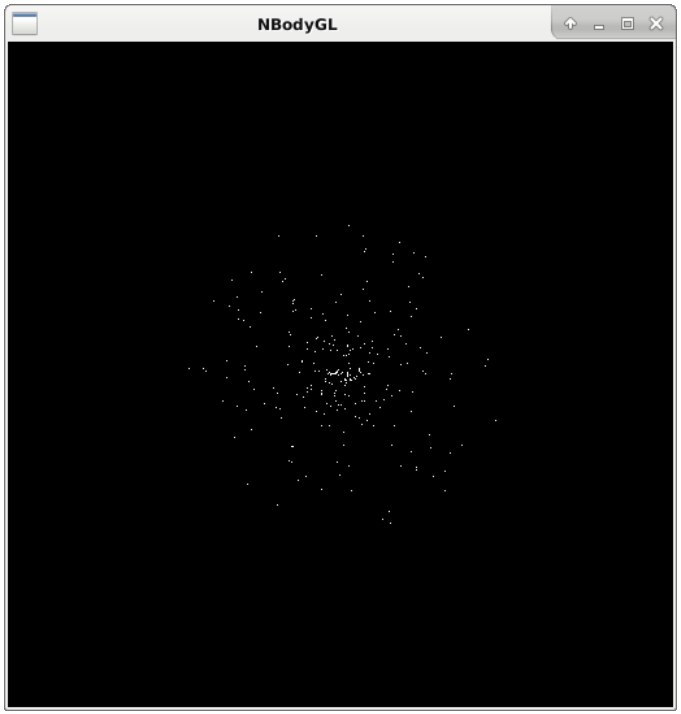


- Pas d'évolution significative pour les (GP)GPUs
- Ratio largement croissant pour les CPU : Intel & AMD

Performances par Watt
Pi Dart Dash / TDP



- L'efficacité des GPU
- La progression des CPU

Une démo visuelle ? Nbody version OpenGL

- En local, 5 « périphériques » OpenCL :
  - 3 implémentations CPU pour Kabylake : AMD, Intel, PortableCL
  - 2 implémentations GPU : Nvidia pour MX150 et Beignet pour Intel HD
- En distant :
  - Tesla P100
  - GTX 1080Ti
  - GTX 1080
  - Vega 64
  - Broadwell
  - Skylake