GPGPU
Implications sur la Sécurité

Lucas Kasey Fernandez

Sommaire

- CPU vs. GPU
- Architectures Massivement Parallèles
- Quelques logiciels et projets
- Quelques chiffres
- Démonstration

INTRODUCTION

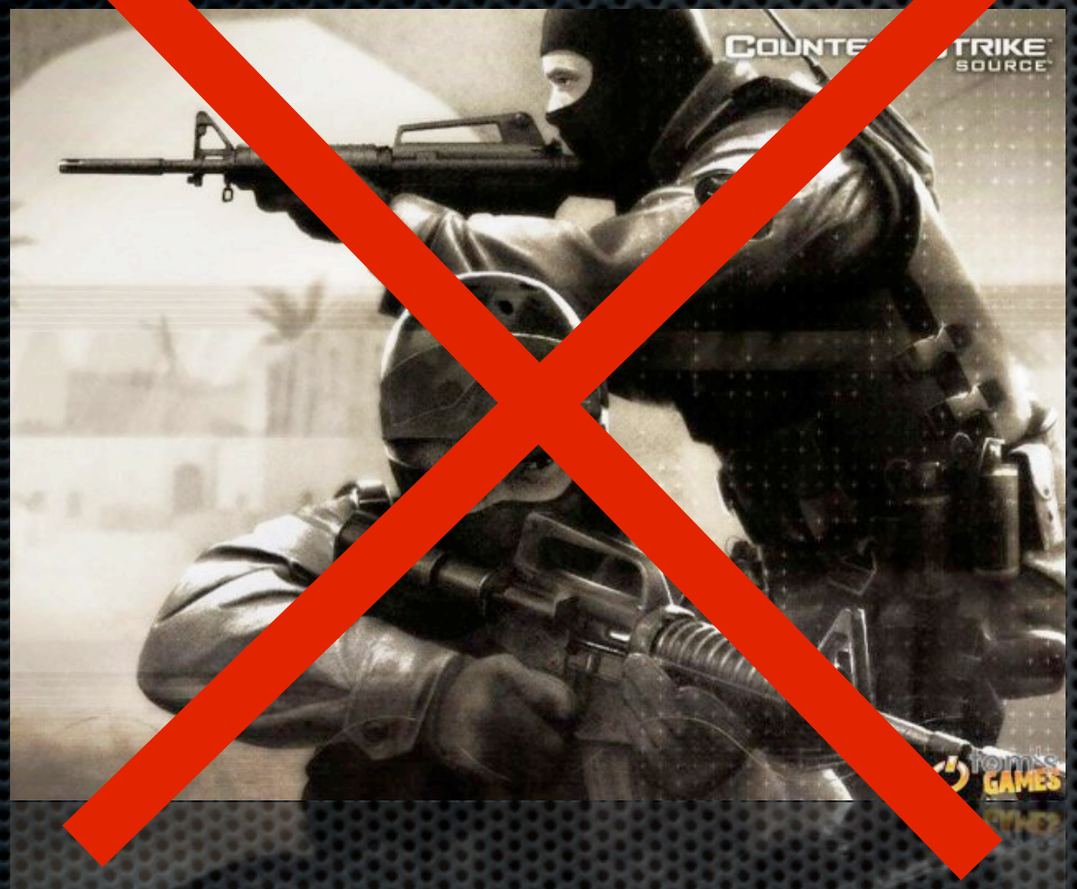
Le GPGPU c'est quoi ?

- Wikipédia : « GPGPU est l'abréviation de **General-Purpose Processing on Graphics Processing Units**
- c'est-à-dire calcul générique sur un processeur graphique »

Le GPGPU c'est quoi ?

- En bref : faire réaliser les calculs initialement destinés aux **processeurs à la carte graphique**

Le GPGPU c'est quoi ?

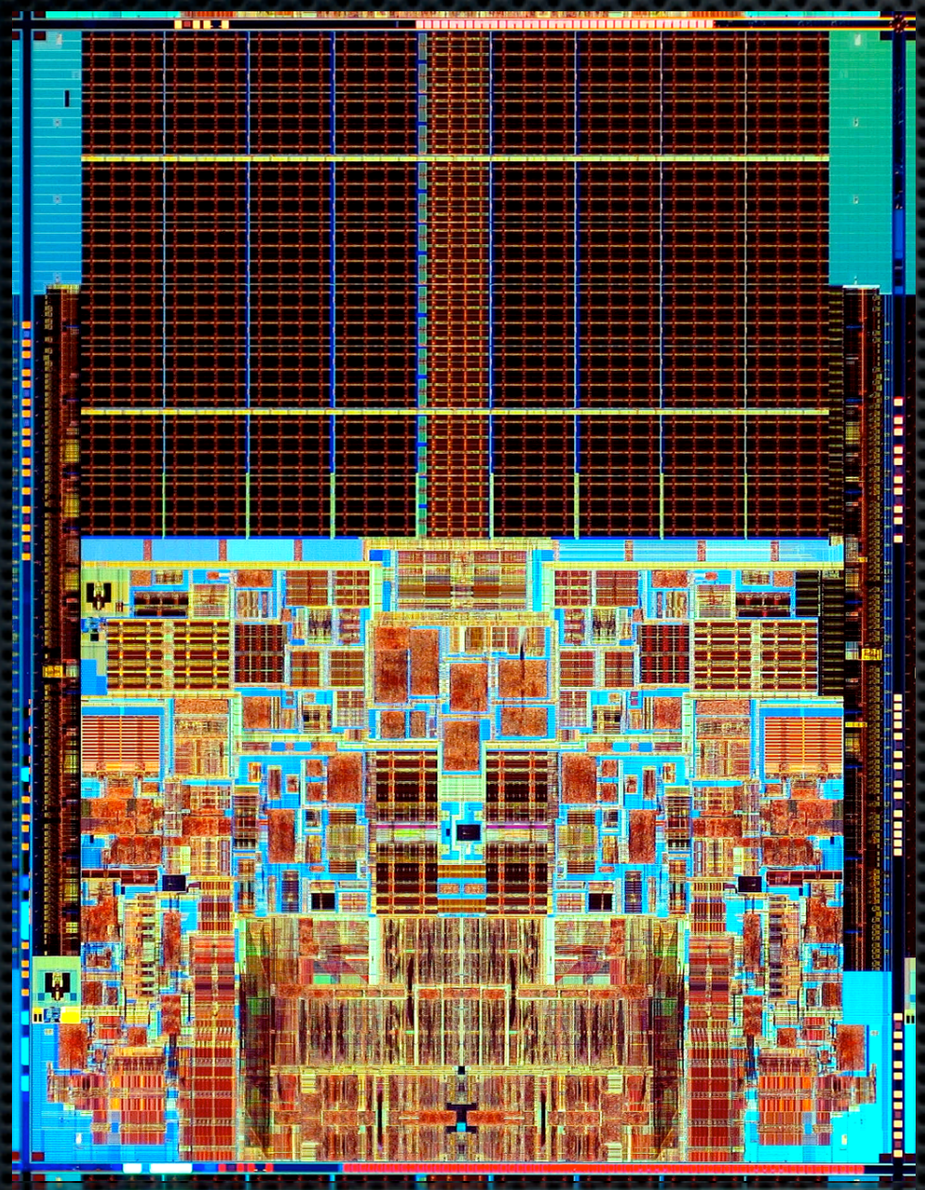


GPU vs. CPU

- CPU vs. GPU
- Architectures Massivement Parallèles
- Quelques logiciels et projets
- Quelques chiffres
- Démonstration

CPU vs. GPU

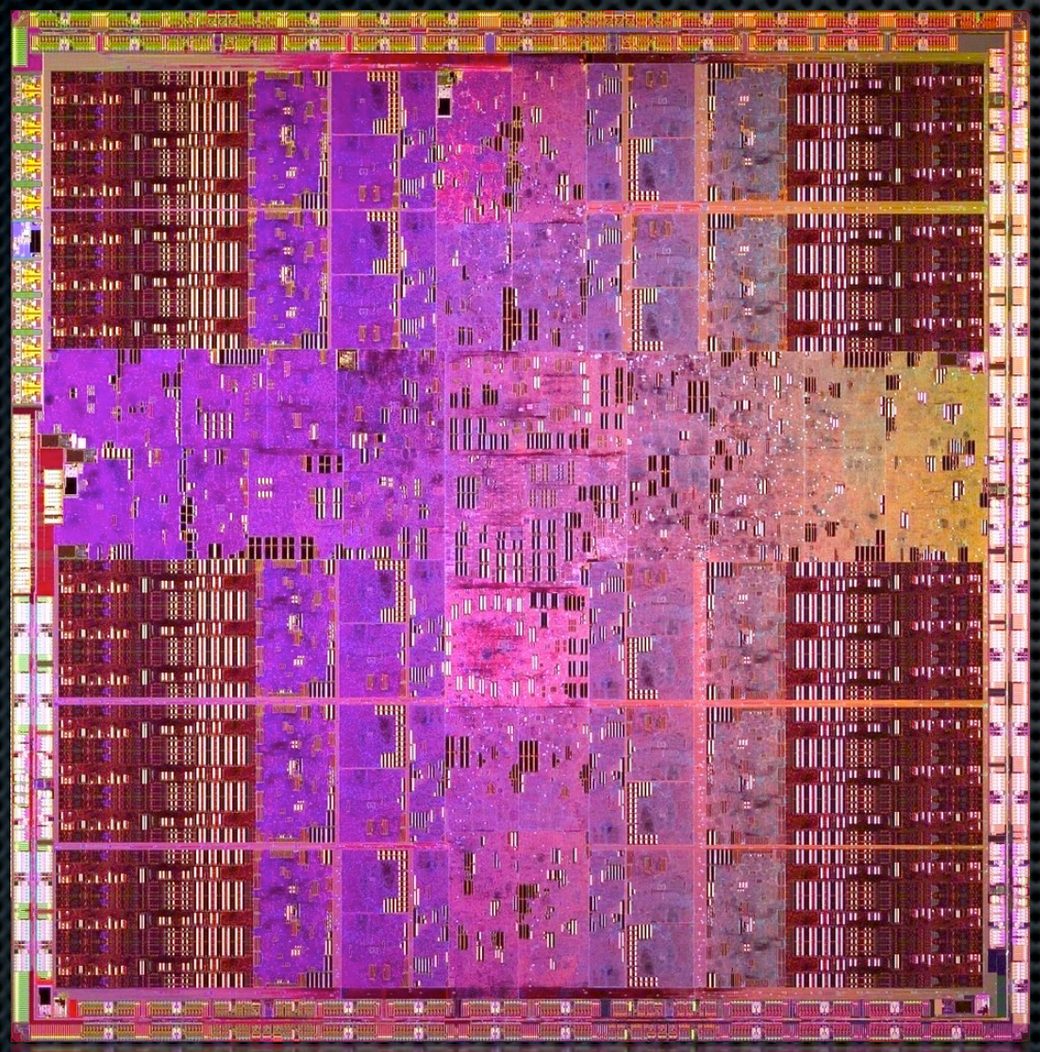
Caractéristiques d'un CPU



Intel Core2Duo

- Intel Core i7-965 :
- 4 Coeurs
- 51 Gf/s
- 230 Watts
- 1000 €

Caractéristiques d'un GPU



NVIDIA GT200

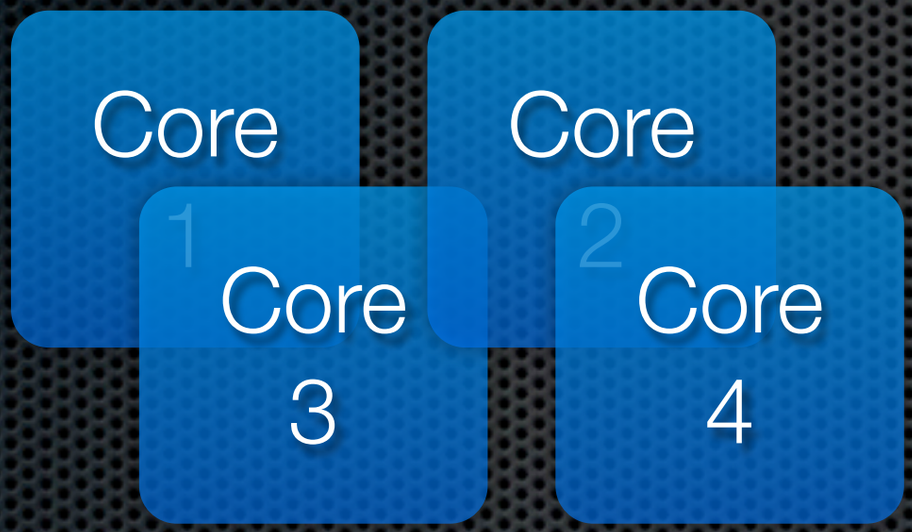
- ATI 5970 :
- 1600 x 2 Coeurs
- 4,6 Tf/s
- 500 Watts
- 550 €

Caractéristiques CPU - GPU

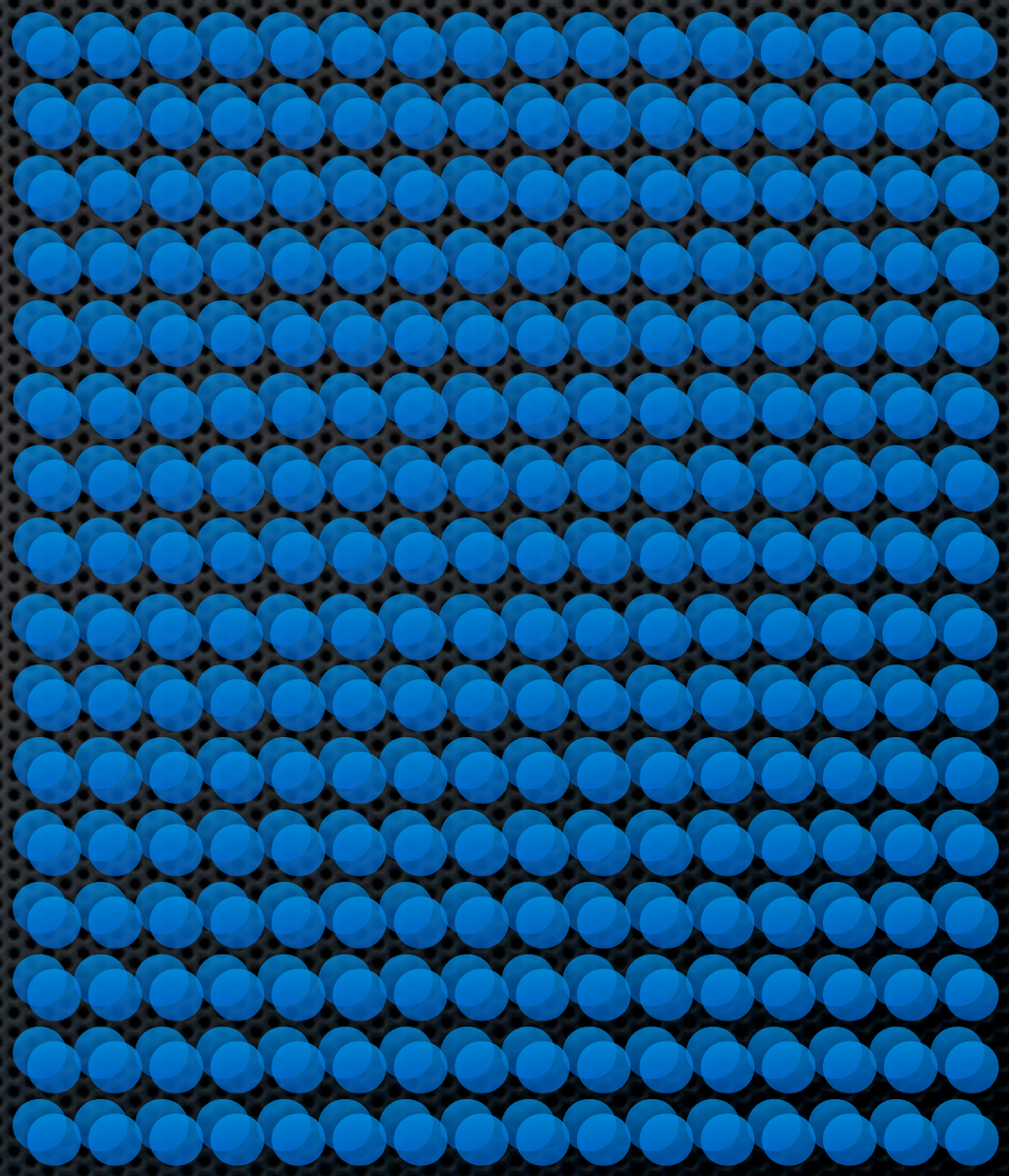
- Les GPUs sont donc :
  - Moins cher
  - Plus puissant
  - Moins gourmand en ressources

Avantages du CPU vs GPU

Avantages du CPU vs GPU



CPU : 4 threads



GPU : 512 threads

Architectures M.P

- GPU vs. CPU
- Architectures Massivement Parallèles
- Quelques logiciels et projets
- Quelques chiffres
- Démonstration

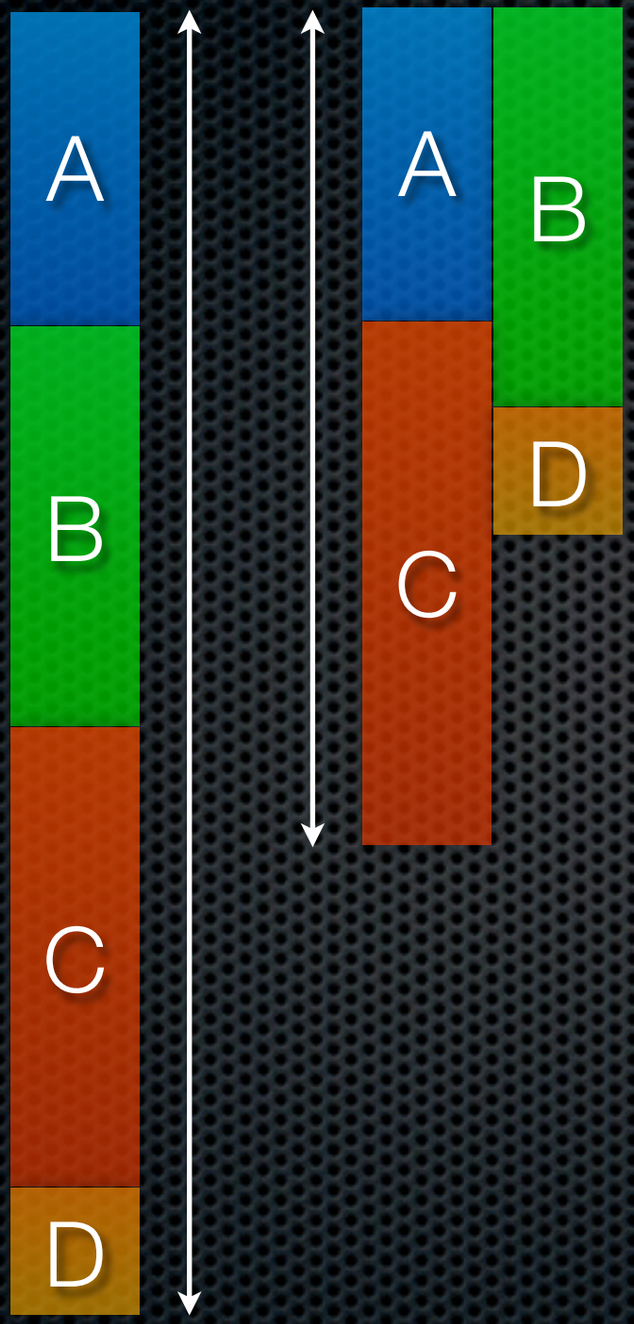
Architectures Massivement Parallèles

Définition

- Le terme de **massivement parallèle** est utilisé lorsque le système comporte de **quelques dizaines de processeurs à plusieurs dizaines de milliers de processeurs**.

X 10 000

Multithread et MPI



- Processus léger
- Ordonnancé au niveau logiciel
- Parallélise les tâches
- Lourd à mettre en place
- MPI norme de calcul parallèle

OpenCL et CUDA

- Frameworks de parallélisation massive

OpenCL

- OpenCL :
  - OpenSource
  - Compatible ATI et NVIDIA
- CUDA :
  - Propriétaire
  - Rejoins OpenCL
  - Compatible NVIDIA Only

NVIDIA®

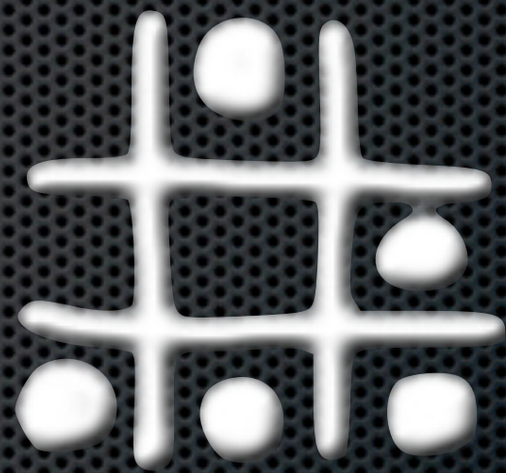
CUDA

Quelques logiciels et projets

- CPU vs GPU
- Architectures Massivement Parallèles
- Quelques logiciels et projets
- Quelques chiffres
- Démonstration

QUELQUES LOGICIELS ET PROJETS

Oclcrack



- Libre
- Implémenté en C++
- Disponible en binaire
- Bon exemple pour commencer la programmation GPGPU
- http://sghctoma.extra.hu/index.php?p=entry&id=11

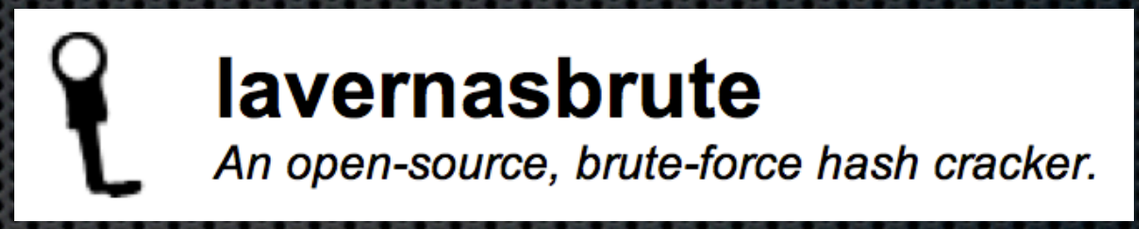
CUDA Multiforcer

- Libre
- Nécessite l'installation des drivers CUDA
- Code dit : «crado»
- Puissant et bien optimisé
- Nécessite quelques améliorations
- http://www.cryptohaze.com/bruteforcers.php

Oclhashcat

- Freeware (code fermé)
- Plus performant que CUDA-Multiforcer
- Très flexible
- Linux Windows (binaires uniquement)
- http://hashcat.net/oclhashcat/

Laverna's brute



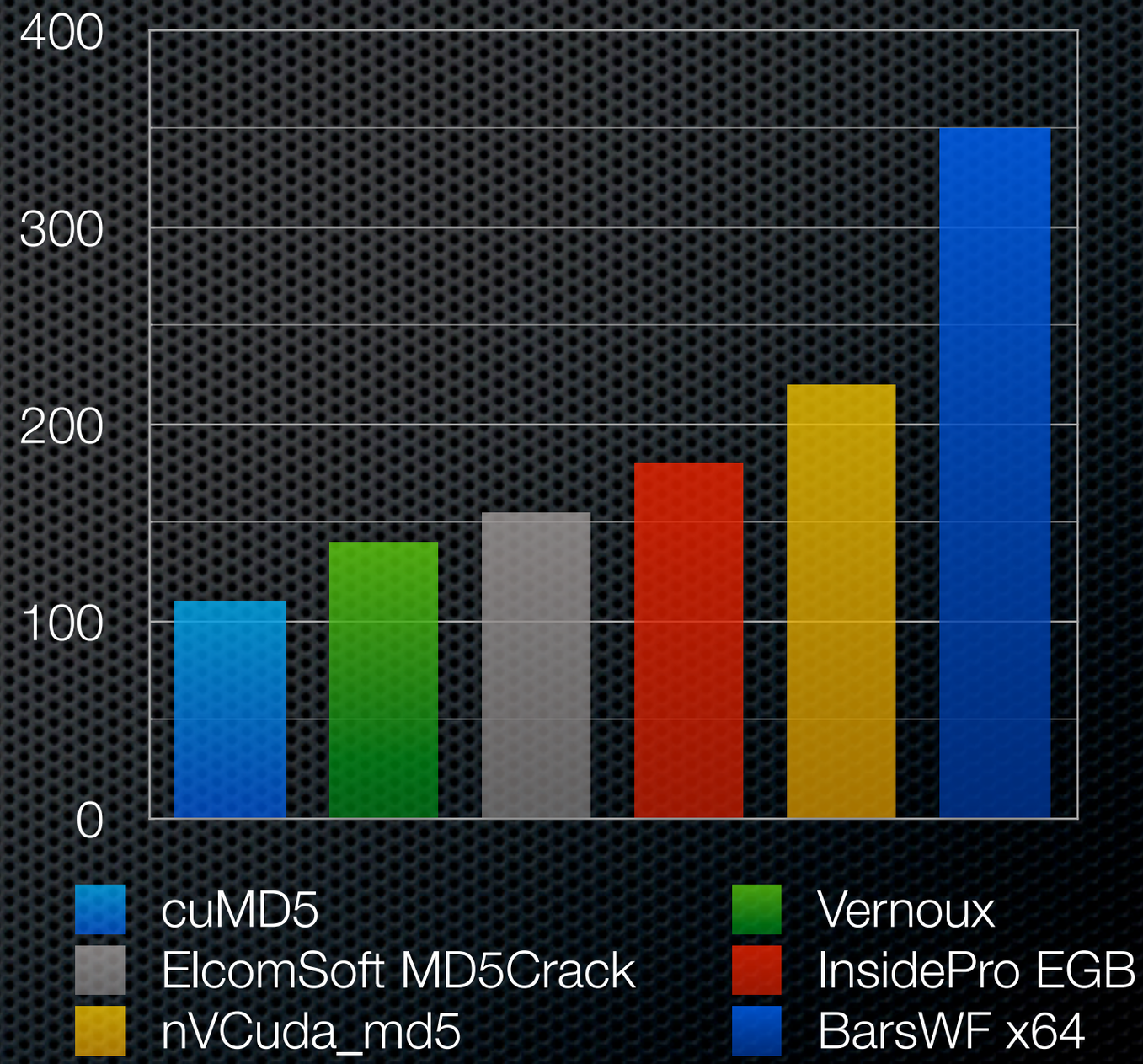
- Librairie de brute force libre et universelle
- Supporte un grand nombre de hash / système crypto
- Implémenté en C++
- Ne supporte pas encore OpenCL ni CUDA
- http://code.google.com/p/lavernasbrute/

Backtrack 4

- Libre
- Propose des tutos GPGPU
- Propose des outils optimisé GPGPU
- Bon travail de diffusion et intégration
- http://www.backtrack-linux.org

BarsFW

- Freeware
- Code source fermé
- Windows only
- Extrêmement rapide
- http://3.14.by/en/md5



Quelques chiffres

- CPU vs GPU
- Architectures Massivement Parallèles
- Quelques logiciels et projets
- Quelques chiffres
- Démonstration

QUELQUES CHIFFRES

Quelques chiffres

Temps de brute-force d'un hash MD5 :

Pass : kkkkkkj
CPU : I7-620M 21GFlop/s
GPU : GT330M 180GFlop/s

Quelques chiffres

Temps de brute-force :

- MD5
- 7 caractères
- Alphanumérique
- 20 minutes max.
- multiplie par 36 par caractère sup.
- 12 heure pour 8 caractères

Démonstration

- CPU vs GPU
- Architectures Massivement Parallèles
- Quelques logiciels et projets
- Quelques chiffres
- Démonstration

DEMONSTRATION

DES QUESTIONS ?¿

Liens

- http://sghctoma.extra.hu/index.php?p=entry&id=11
- http://www.bindshell.net/tools/johntheripper
- http://www.khronos.org/opencl/
- http://www.nvidia.fr/object/cuda_home_new_fr.html
- http://kasey.fr
- http://bearstech.com