

GPGPU et OpenCL



OpenCL

Sommaire

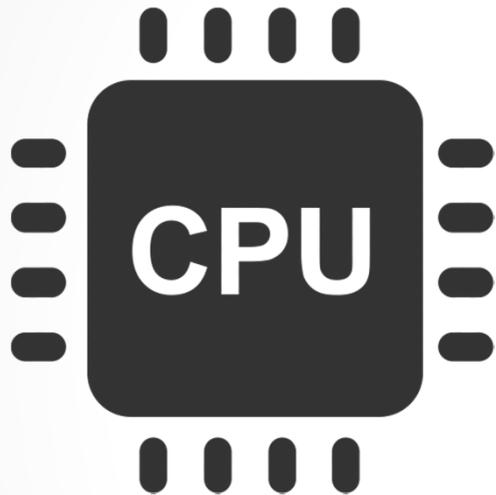
1- GPGPU

GPU, calcul parallèle, coût et avantages

2- OpenCL

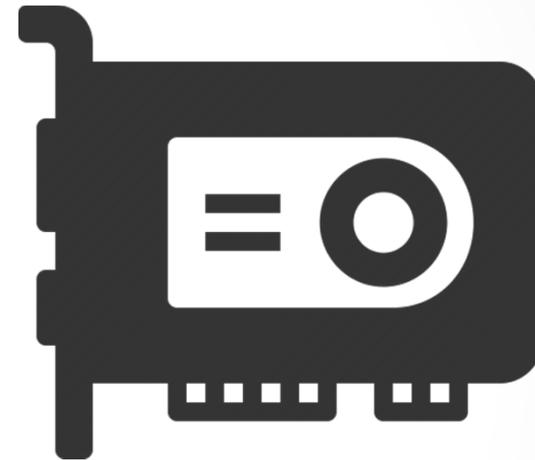
OpenCL, CUDA, DirectCompute

I - GPGPU



Unité de calcul général

Un gros cœur

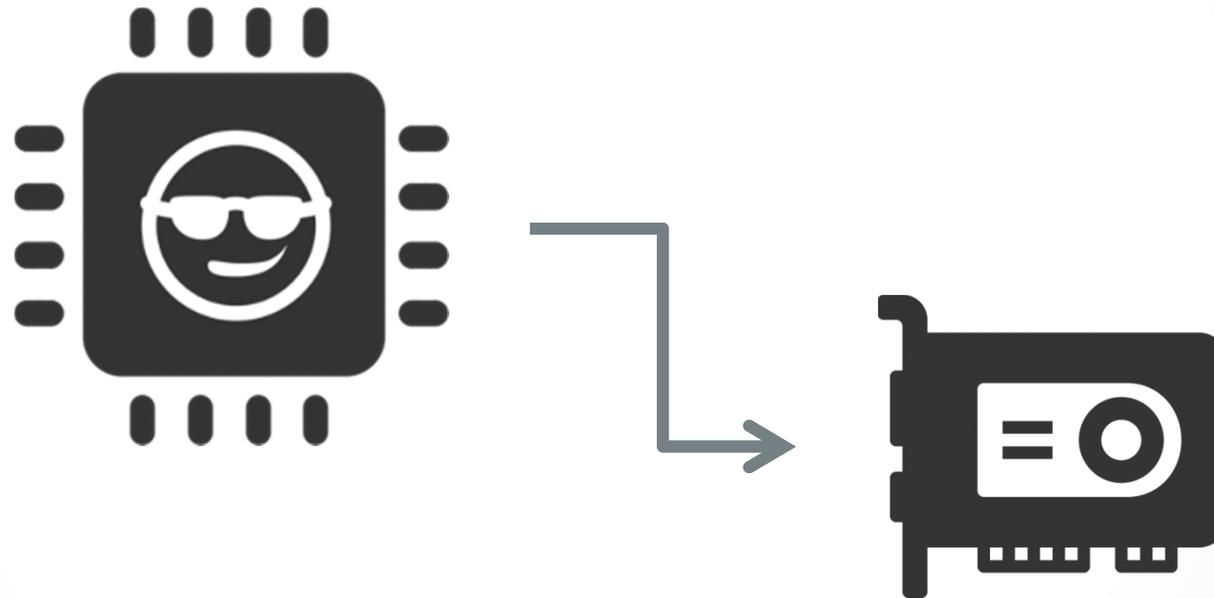


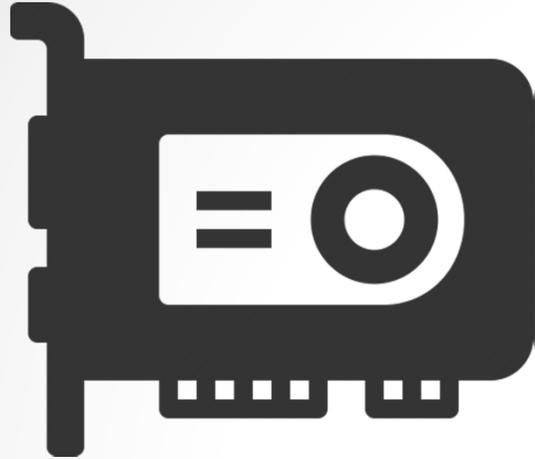
Unité de calcul spécialisé

Plein de petits cœurs

GPGPU

General-Purpose computing on Graphics Processing Units





1999 **Nvidia GeForce256**
« world's first GPU »

OpenGL & DirectX

2001 **GPGPU**
GPU à la place du CPU

2007 **CUDA**
API GPGPU de Nvidia

2009 **OpenCL**
OpenAPI GPGPU



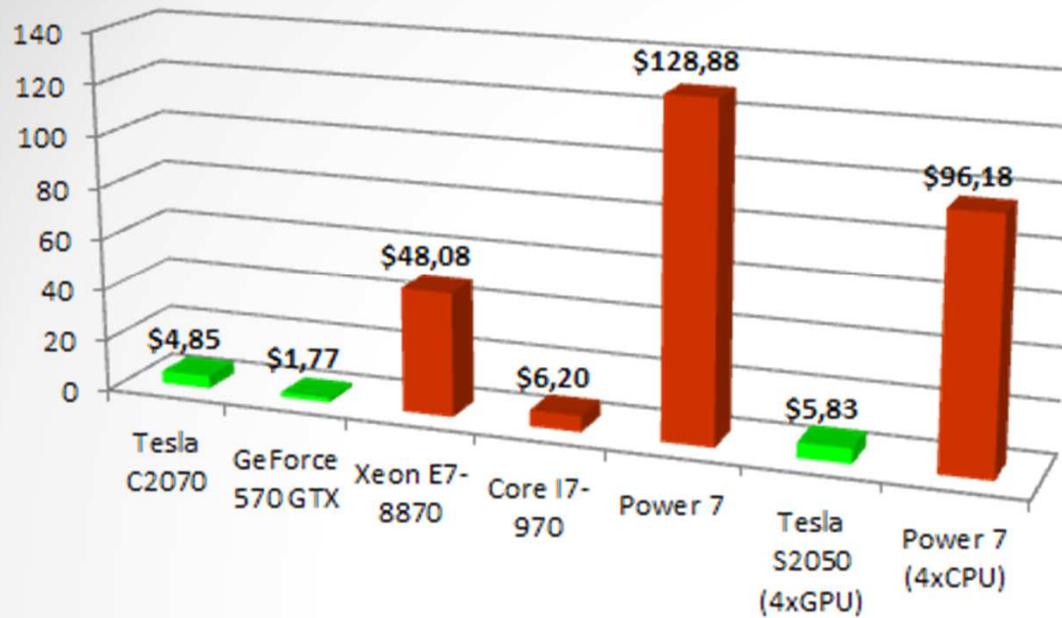
Utilisation de CUDA dans SciFinance
Modèles 20-100x plus rapide



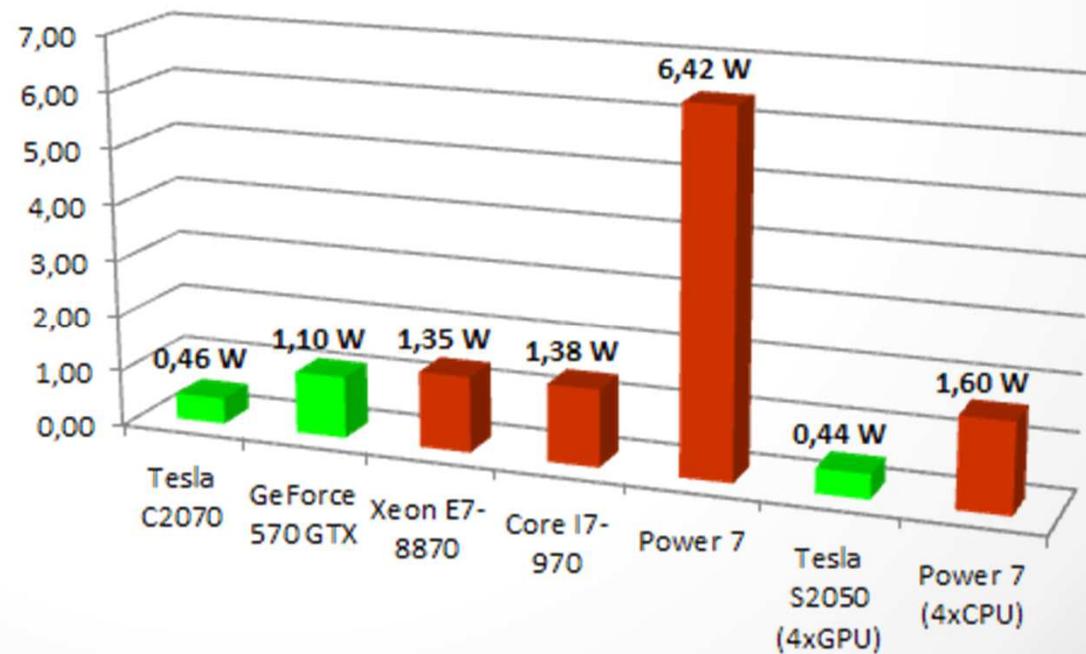
BNP PARIBAS

Avant : 500 CPU (25kw)
Après : 2 Tesla S1070 (2kw)
Temps de réponse divisés par 15

Coût d'un GFlop (en \$)



Watt consommé par GFlop



2- OpenCL

Framework **standardisé ouvert** pour des infrastructures hétérogènes

Initialement développé par **Apple**

puis confié à **Khronos Group** en collaboration avec **AMD, IBM, Qualcomm, Intel** et **Nvidia**

10 Décembre 2008 :

1^{ère} démonstration d'OpenCL par **AMD** et **Nvidia**

28 Août 2009 : OpenCL 1.0

intégré dans Mac OS X 10.6

Implémentations des constructeurs :

Altera

AMD

Apple

ARM Holdings

Creative Technology

IBM

Imagination Technologies

Intel

Nvidia

Qualcomm

Samsung

Vivante

Xilinx

ZiiLABS

Langage OpenCL : basé sur C99 (adapté)

```
// Multiplies A*x, leaving the result in y.  
// A is a row-major matrix, meaning the (i,j) element is at A[i*ncols+j].  
__kernel void matvec(__global const float *A, __global const float *x,  
                    uint ncols, __global float *y)  
{  
    size_t i = get_global_id(0);  
    __global float const *a = &A[i*ncols];  
    float sum = 0.f;  
    for (size_t j = 0; j < ncols; j++) {  
        sum += a[j] * x[j];  
    }  
    y[i] = sum;  
}
```

Appelé depuis un programme hôte

Parallélisme de tâches et de données
Abstraction pour le programmeur

Versions

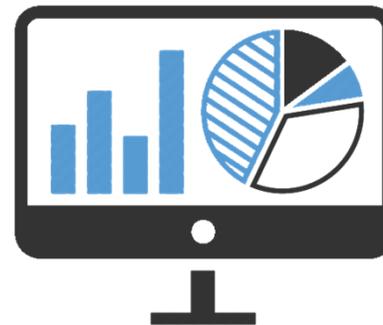
OpenCL 1.0	2009
OpenCL 1.1	2010
OpenCL 1.2	2011
OpenCL 2.0	2013
OpenCL 2.1	2015
OpenCL++ (C++14)	
Vulkan	
DirectX12	
OpenCL 2.2	2016

Monitoring



Debugging

Outils



Merci de votre attention !

Sources

<https://www.khronos.org/opencv/>

<https://en.wikipedia.org/wiki/OpenCL>

<https://en.wikipedia.org/wiki/GPGPU>

<http://blog.octo.com/>