

Programmation et calcul générique sur GPU

Sommaire

1. Qu'est-ce que la « programmation GPU » ?
2. Petite histoire ...
3. Les différentes bibliothèques
4. OpenCL

General-purpose computing on graphics processing units (GPGPU)

- Usage du processeur graphique (GPU) pour effectuer des calculs normalement réalisés par le processeur.
- Permet de faire des calculs en parallèles.

Motivations

- Calculs complexe sur un volume important de données en exploitant le GPU.
- Exemple :
 - Multiplication matricielle et calcul sur des vecteurs en N dimensions

Petite histoire

- 2001 : Multiplication matricielle exploitant le GPU
- 2005 : Décomposition LU sur GPU (plus rapide que sur CPU)
- 2006 ? : Sortie de CUDA
- 2009 : OpenCL 1.0

Différentes bibliothèques

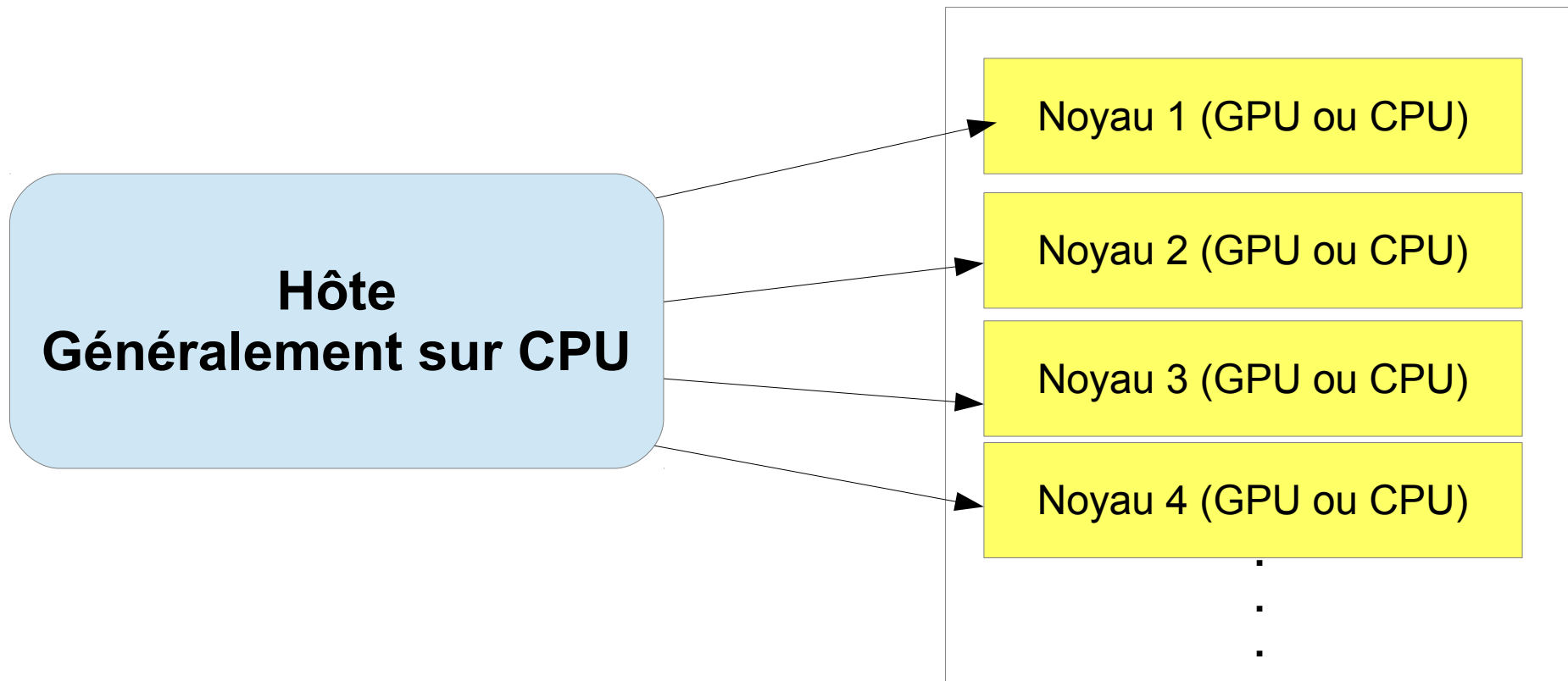
- DirectCompute (Microsoft)
- Compute Unified Data Architecture (CUDA) (Nvidia)
- OpenACC
- Open Compute Library (OpenCL) (Khronos)

OpenCL

- Bibliothèque de calcul GPGPU pour des systèmes parallèles hétérogènes (~ CPU multicoeurs + GPU)
- Basé sur un modèle à l'intersection entre le monde CPU et le monde GPU.
- Intégrée dans Vulkan

OpenCL – Modèle de programmation - 1

- Distinction processeur hôte/périphériques (CPU, GPU)



OpenCL – Modèle de programmation - 2

- Parallélisme de tâches
 - Multicoeurs (CPU/GPU)
- Parallélisme de données
 - SPMD (Single Program Multiple Data)
 - SIMD (Single Instruction Multiple Data)

OpenCL VS CUDA

- CUDA
 - Très orienté calcul GPU
 - Débogage aisé (NVIDIA Parallel Sight)
- OpenCL
 - Pas exclusivement dédié au calcul GPU
 - Calcul à haute performance (HPC)
 - Plus dure à déboguer

References

- Calcul sur GPU
- CUDA : <https://developer.nvidia.com/cuda-zone>
- OpenACC : <https://www.openacc.org/>
- OpenCL : <https://www.khronos.org/opencl/>