

1 Dépoussiérage

1.1 Branchez le Jetson TX2

N'oubliez rien : alimentation, HDMI-DVI, clavier souris USB, antennes pour le WiFi ou câble ethernet, ...

Allumez-le. Indice : la carte possède un bouton power.

1.2 Récupérez des infos techniques sur le Jetson TX2

Utilisez des commandes comme :

- | | |
|----------------------------------|----------------------------------|
| • <code>lspci</code> | • <code>cat /proc/meminfo</code> |
| • <code>lscpu</code> | • <code>glxinfo</code> |
| • <code>lshw</code> | • <code>clinfo</code> |
| • <code>dmidecode</code> | • ... |
| • <code>cat /proc/cpuinfo</code> | |

1.3 Architecture du SoC

- Quelle est l'architecture du Jetson TX2 ?
- Comment la mémoire est-elle organisée ? Entre le CPU et le GPU ?
- Quelle est l'architecture du GPU (Fermi, Kepler, Pascal, ...) ?

1.4 CUDA

- Quelle version de CUDA est installée ?
- Quelle version du driver Nvidia ?
- Combien de coeurs CUDA ? Combien de coeurs CPU ?
- Combien de SM (streaming multiprocessor) ?
- Limites en taille de la grid et des blocks de threads ?

Utilisez des commandes comme :

- `nvidia-smi`
- `lsmod`
- `modinfo`
- `ls /sys/module/nv*`
- `deviceQuery`
- ...

2 Programmation GPU

2.1 Transferts mémoire

Avec les exemples du cours, écrivez un programme C/C++ (ou Python) permettant de mesurer la bande passante mémoire entre le CPU et le GPU.

Utilisez la commande `time` pour mesurer le temps d'exécution de vos programmes.

Est-ce cohérent par rapport à la documentation technique ?

2.2 Addition de vecteurs

Avec les exemples du cours, écrivez un programme C/C++ (ou Python) effectuant l'addition de deux vecteurs de 10000 float dans un troisième vecteur. Vous écrirez une version qui n'utilise que le CPU, et une version qui lance le calcul sur le GPU Nvidia en CUDA.

Mesurez les temps d'exécution de vos versions CPU et GPU.

Que constatez-vous ? Comment l'expliquez-vous ?

Modifiez vos programmes de manière à ce que la densité de calcul float augmente, par exemple faites faire davantage de calculs sur les deux float d'entrée (pas seulement une addition) avant de charger le résultat dans le troisième float.

Refaites diverses mesures de temps. Que constatez-vous ?

Rendez votre programme capable de lire le nombre d'éléments à traiter sur la ligne de commande.

- Combien de floats peuvent être additionnés en parallèle sur le GPU ?
- Quelle configuration de grille et blocks avez-vous choisi ?
- Quelle taille maximum de vecteur de float puis-je ainsi additionner sur le Jetson TX2 en CUDA ?

2.3 CUDA SDK

Explorez les exemples CUDA se trouvant dans

`/usr/local/cuda/samples/0_Simple`

et

`/usr/local/cuda/samples/1_Uutilities`

Compilez et testez `vectorAdd` et `bandwidthTest`.