



Forum: Questions & Réponses

Topic: Objet-Relations-Calques (Object-Relations-Layers)

Subject: Re: Objet-Relations-Calques (Object-Relations-Layers)

Posté par: LadeHeria

Contribution le : 16/1/2015 8:31:10

Hum, pas vraiment, la manière d'envisager les choses est plutôt la suivante : la carte graphique, c'est comme un mini système à part entière, mais tout intégré. Un "processeur" (graphique), une carte mère, des puces mémoires, contrôleur mémoire, contrôleur vidéo, etc... Système qui va communiquer avec un autre : processeur, carte mère, mémoire RAM, etc. Et qui jusque à peu lui était totalement asservi (il est toujours en grande partie, c'est le processeur qui donne les ordres).

Donc le processeur graphique va communiquer à une certaine vitesse avec sa mémoire graphique (GDDR5), le processeur de calcul à une certaine vitesse avec sa RAM, DDR3/DDR4 (30/60 ou plus Go/s) (sachant qu'il y a différents niveaux de cache; L1, L2 etc qui "vont très vite", en fait il faut comprendre qu'à peu près tout les accès mémoires "tuent" complètement la vitesse de calcul. Pour commencer à accéder des données dans le cache L1, tu vas y passer 2-4 cycles d'horloge, L2 10, L3 20-30, etc RAM 100 ns ou plus (équivalent à 300 cycles pour un processeur à 3Ghz).

Alors que quand tu veux faire communiquer la carte graphique et la RAM ou une autre carte graphique, tu passes par un port PCI, qui lui est beaucoup plus limité en bande passante (le lien processeur-RAM est beaucoup plus rapide).

Et en plus, quand tu veux accéder à une donnée dans la RAM depuis la CG, tu dois passer par le contrôleur mémoire de la RAM. Donc repasser par le processeur pour que le driver dise au contrôleur mémoire d'écrire (sur la CM ou le processeur), et donc perdre du tps, etc.

Donc non, on ne peut pas considérer que la RAM est une carte fille.

@Dans ton poste 37, tu dis "c'est possible en plus de la partager; en OpenCL en tout cas...". Peux-tu expliquer un-peu-plus ce détail avec la perte de performance en occurrence, merci d'avance.

Il y a des instructions qui existent en OpenCL/CUDA pour écrire/lire des données en RAM, je pense que l'explication au dessus permet de comprendre pourquoi c'est plus lent que d'aller piocher dans la GRAM (250 GB/S vs 16 pour un port PCI, latence qui explose), à noter qu'il semblerait que les nouvelles versions évitent le passage par le processeur (le processeur exécute le driver) en permettant d'accéder directement au contrôleur mémoire (qui maintenant sur les nouvelles gammes Intel est intégré sur le "processeur"), donc on économise en latence et en occupation CPU.

Désolé, mon post est vraiment bordélique.

LadeHeria