



Open CL

KMS - Martin Dubois, ing

2015-09-16

KMS - Martin Dubois, ing.

1



Plan de la présentation

- Présentation du formateur et des participants
- Pourquoi OpenCL?
- *Host vs Kernel*
- Les modèles proposés par OpenCL
- Obtenir et installer un environnement
- Programmation du côté *host* — Première partie
- Programmation des *kernels* — Première partie
- Programmation du côté *host* — Seconde partie
- Programmation des *kernels* — Seconde partie
- Bonnes pratiques générales
- Technique de déverminage
- Les autres cours offerts

2015-09-16

KMS - Martin Dubois, ing.

2

Partie 1

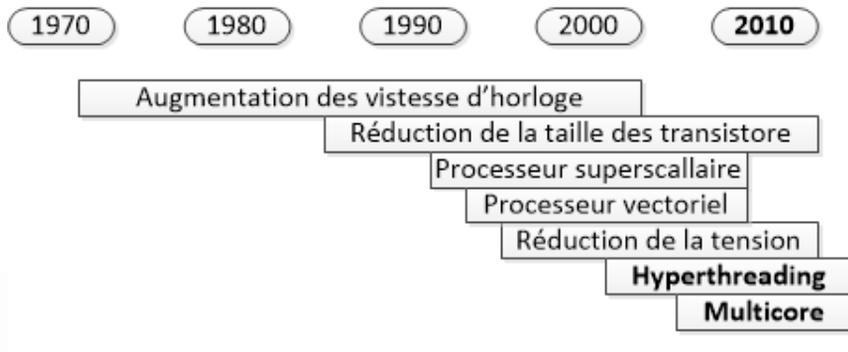
POURQUOI OPENCL?

2015-09-16

KMS - Martin Dubois, ing.

3

Un peu d'histoire!



2015-09-16

KMS - Martin Dubois, ing.

4

Maintenant!

- Il n'est désormais pas rare de voir un ordinateur avec 12 ou 16 **cores** logiques, voir plus!
- Et si nous pensons au **GPU**, ce sont des centaines de **cores** qui sont à notre disposition!
- Les développeurs de logiciels aimeraient bien exploiter cette puissance, mais la programmation « **parallèle** » n'est pas simple!

Les défis de la programmation parallèle

- Il faut, au niveau de l'algorithme, identifier des éléments qu'il est possible de paralléliser
 - Au niveau des « **données** » à traiter
 - Au niveau des « **tâches** » à exécuter
- Il faut identifier les **dépendances** entre ces éléments
- Il faut mettre en place un logiciel gérant la **communication** et la **synchronisation** entre les éléments

C'est encore pire avec les *GPUs*

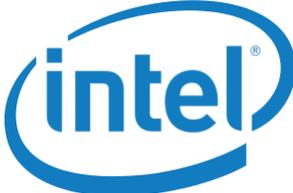
- Les **GPUs** ont leur propre mémoire et il faut donc transférer les données.
- Il faut utiliser du code spécifiquement compilé pour ces **GPUs**.
- Les architectures sont complexes et différentes d'un vendeur à un autre.
- Le grand nombre de **cores** met beaucoup de pression sur la **synchronisation**.

2015-09-16

KMS - Martin Dubois, ing.

7

 OpenCL — Participants  

2015-09-16 KMS - Martin Dubois, ing. 8

Un standard intelligent!

- **OpenCL** ne réinvente pas un langage de programmation, il est basé sur **C99**
- Naturellement, il y a quelques ajouts, mais la plupart prennent la forme de fonctions « supplémentaire »

Partie 2

HOST VS KERNEL



Un peu de vocabulaire!

- Un programme **OpenCL** se divise toujours en deux parties : la partie **host** et le ou les **kernels**
- La partie **host** est une application normale écrite en **C/C++** et utilisant l'API de **OpenCL** pour contrôler l'exécution du ou des **kernels**
- Les **kernels** sont les fonctionnalités écrites en « **OpenCL C** » et s'exécutant sur le ou les processeurs hétérogènes.

2015-09-16

KMS - Martin Dubois, ing.

11



Un peu de vocabulaire!

Host

- Écrit en **C/C++** ou même dans d'autres langages
- Compilé en utilisant votre compilateur favori
- Exécuté sur l'ordinateur principal comme n'importe quelles autres applications

Kernel

- Écrit en **OpenCL C**
- Compilé en utilisant le compilateur fourni par le fabricant du processeur hétérogène
- Exécuté sur :
 - Le processeur
 - Un *DSP*
 - Un *FPGA*
 - Un *GPU*

2015-09-16

KMS - Martin Dubois, ing.

12

Partie 3

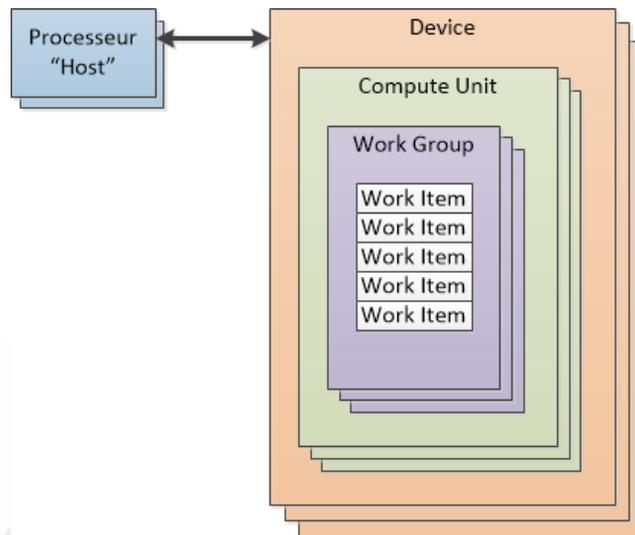
LES MODÈLES PROPOSÉS PAR OPENCL

2015-09-16

KMS - Martin Dubois, ing.

13

Le modèle « matériel »



2015-09-16

KMS - Martin Dubois, ing.

14

