

Introduction à la Programmation Parallèle

Partie 6 - Plateformes de programmation parallèles



Dr Babacar Diop

Département d'Informatique

UFR des Sciences Appliquées et de Technologies

Université Gaston Berger de Saint-Louis

Année scolaire: 2019/2020

Langage de programmation

- Occam
- CUDA
- OpenCL
- MPI

Occam

- Occam est un **langage de programmation concurrent** qui s'appuie sur la logique de **processus séquentiels communicants** (CSP)[1].
- Nommé d'après **William Ockham** du célèbre *Rasoir d'Occam*.
- Occam est un **langage de procédure impérative** (comme Pascal).
- Il a été développé par David May et d'autres à l'INMOS, conseillé par **Tony Hoare**, comme langage de programmation native pour les microprocesseurs de **transputer**, mais des mises en œuvre pour d'autres plates-formes sont disponibles.
- La version la plus connue est **occam 2**

Occam

- La **communication entre processus** fonctionne à travers les canaux appelés « **channels** ».
- Un processus produit les données vers un canal via le symbole "!" Tandis qu'un autre "déetecte" les données avec "?".
- L'**entrée** et la **sortie** ne peuvent se poursuivre que lorsque l'autre extrémité est prête à accepter ou à transmettre des données. Au cas contraire, le processus «bloque» sur le canal.

Occam

Exemples: Soit c est une variable

```
keyboard ? c
```

```
screen ! c
```

```
SEQ
```

```
  x := x + 1  
  y := x * x
```

```
PAR
```

```
  p()  
  q()
```

Occam (Transputers)



Inmos T212, PREQUAL

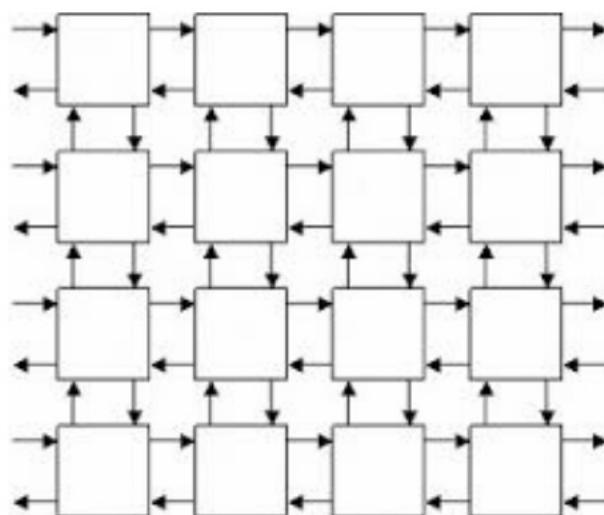
Inmos T222, PREQUAL

STMicroelectronics
IMST225 (Inmos T225).

Inmos T400

Inmos T414

Inmos T425

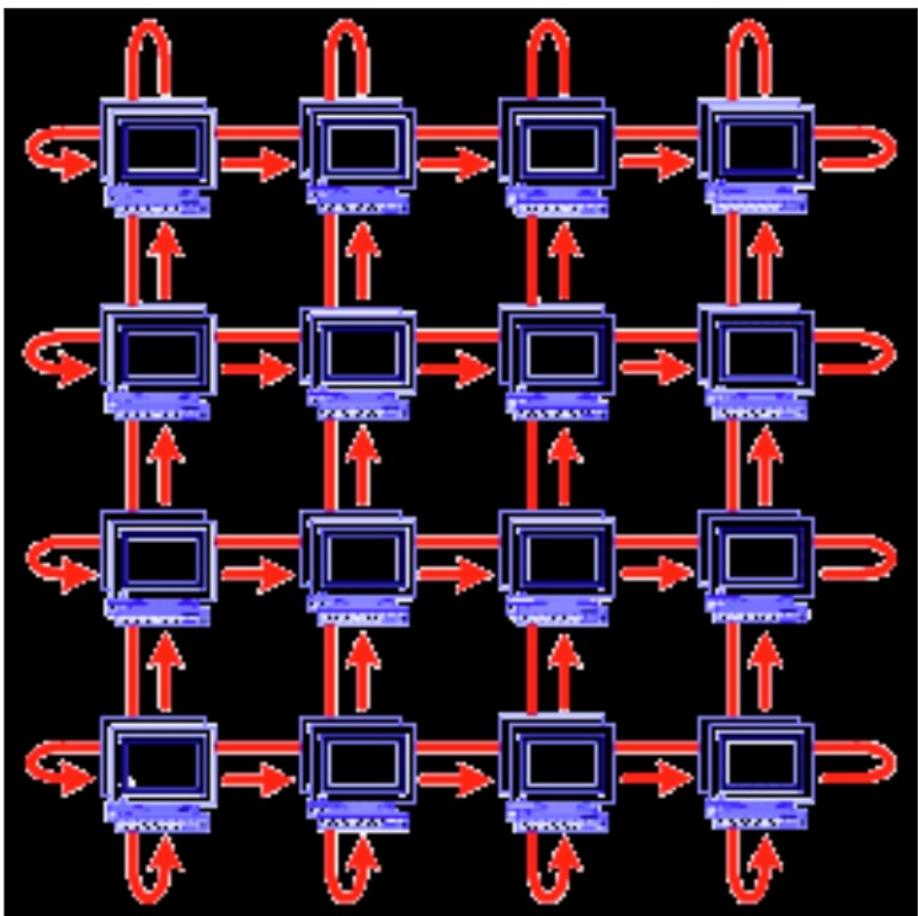


Les **Transputers** étaient une série de microprocesseurs pionniers des années 1980, avec des **mémoire** et des **liens de communication intégrées**, destinées à l'informatique parallèle (Occam).

Ils ont été conçu et produit par **Inmos**, une société de semi-conducteurs basée à Bristol, au Royaume-Uni.

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

Occam : exemples



CUDA : Qu'est-ce que CUDA?



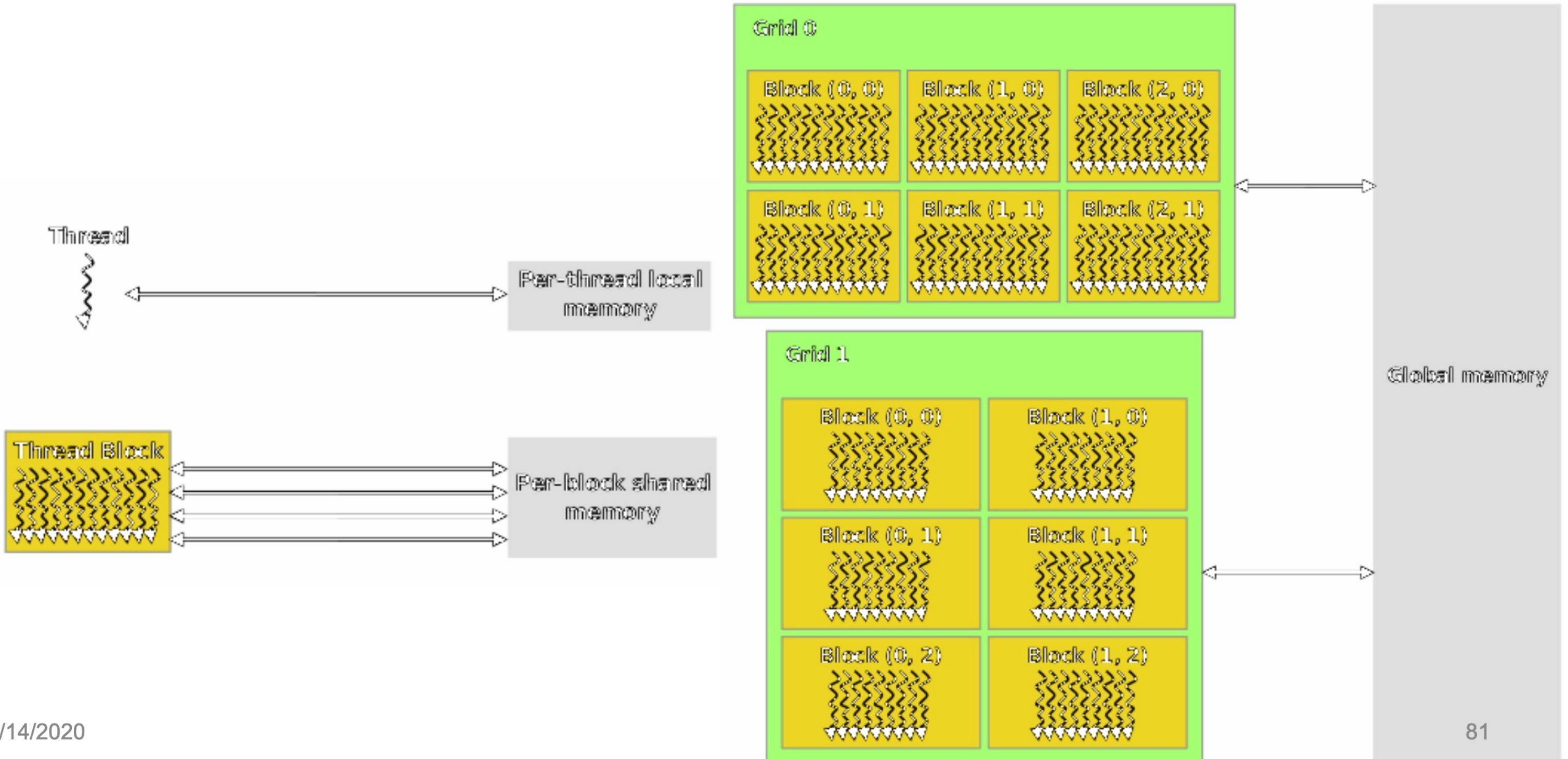
- Une **plate-forme informatique parallèle** et un modèle d'interface de programmation d'applications (**APIs**) créé par **Nvidia**.
- Permet aux développeurs et aux ingénieurs logiciels d'utiliser une **unité de traitement graphique** (GPU) pour un **augmenter les performances d'un programme**.
- Mobilise le **processeur** (CPU) et la **carte graphique** (GPU) pour accélérer l'exécution d'un algorithme parallèle.

CUDA

- **GPU** ? L'unité de traitement graphique (GPU), en tant que processeur spécialisé, répond aux exigences des tâches en 3D à haute résolution et à des tâches gourmandes en calcul.
- **Contexte?** Depuis 2012, les GPU sont devenus des **systèmes multi-core** massivement parallèles, et permettant une manipulation très efficace de gros blocs de données en mémoire.
- **GPU vs CPU** ? Plus efficace que l'unité de traitement central (CPUs). A privilégier pour des algorithmes incluant le traitement de gros blocs de données en mémoire.



CUDA : gestion de mémoire





Flux de traitement simple

1. Copier les données d'entrée de la mémoire du processeur (CPU) vers la mémoire graphique (GPU)
2. Charger le programme GPU et exécuter, mettre en cache les données pour optimiser les performances
3. Copier les résultats de la mémoire GPU vers la mémoire CPU



Zoho Show

cours-algo-parallele-2018.pdf

This file was created using Zoho Show

A comprehensive presentation app that helps you create, collaborate,
present and share slides on the go.

<https://show.zoho.com>