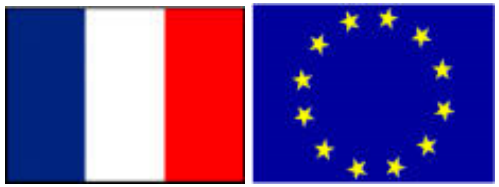


GP-GPU

TD1: Dense matrix product on GPU using registers

Stéphane Vialle

CentraleSupélec université PARIS-SACLAY ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC) LISN LABORATOIRE INTERDISCIPLINAIRE DES SCIENCES DU NUMÉRIQUE

Stephane.Vialle@centralesupelec.fr
http://www.metz.supelec.fr/~vialle

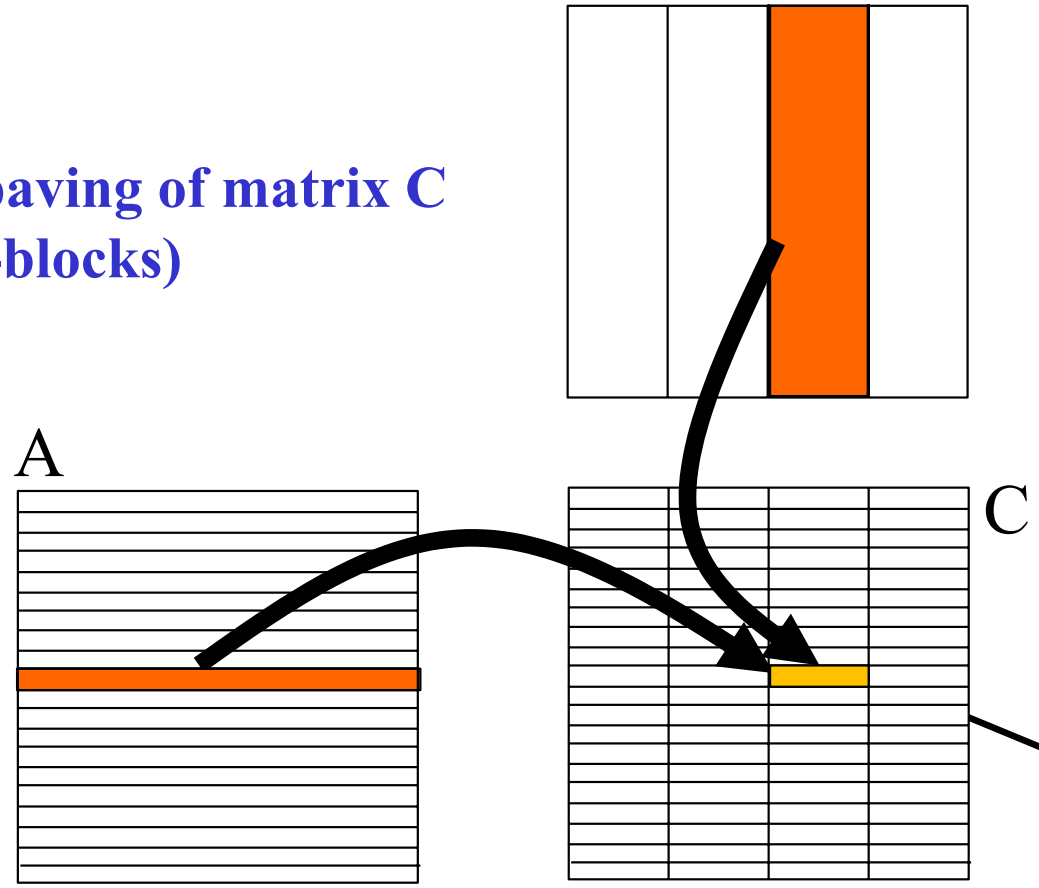
Ex1: 2D-grid of 1D-blocks

Product of matrices of *n*×*n* elements

- 1D-blocks of B_x threads (kernel k0)
- With: $n = k.B_x$ $(k \in \mathbb{N})$

1 - Design the paving of matrix C (2D-grid of 1D-blocks)

2 - Implement *coalescent* accesses to the global memory



POLYTECH PARIS-SACLAY

Ex1: 2D-grid of 1D-blocks

Q1.1 - Definition of the 2D-grid of 1D-blocks

Q1.2 - Definition of the computing kernel

Q1.3 - Analysis of the coalescence

Q1.4 - Number of global memory accesses (not including cache memory):

- Compute the total number of memory accesses <u>requested</u> by the threads

$$N_{RAM\ accesses}^{requested\ by\ all\ threads} = (n_{threads} . n_{RAM\ accesses}^{requested\ by\ 1\ thread})$$

- Compute the total number of memory accesses <u>achieved</u> by the warps

  When accesses are coalescent: 1 warp accesses 32 data in $t_{1\ RAM\ access}$

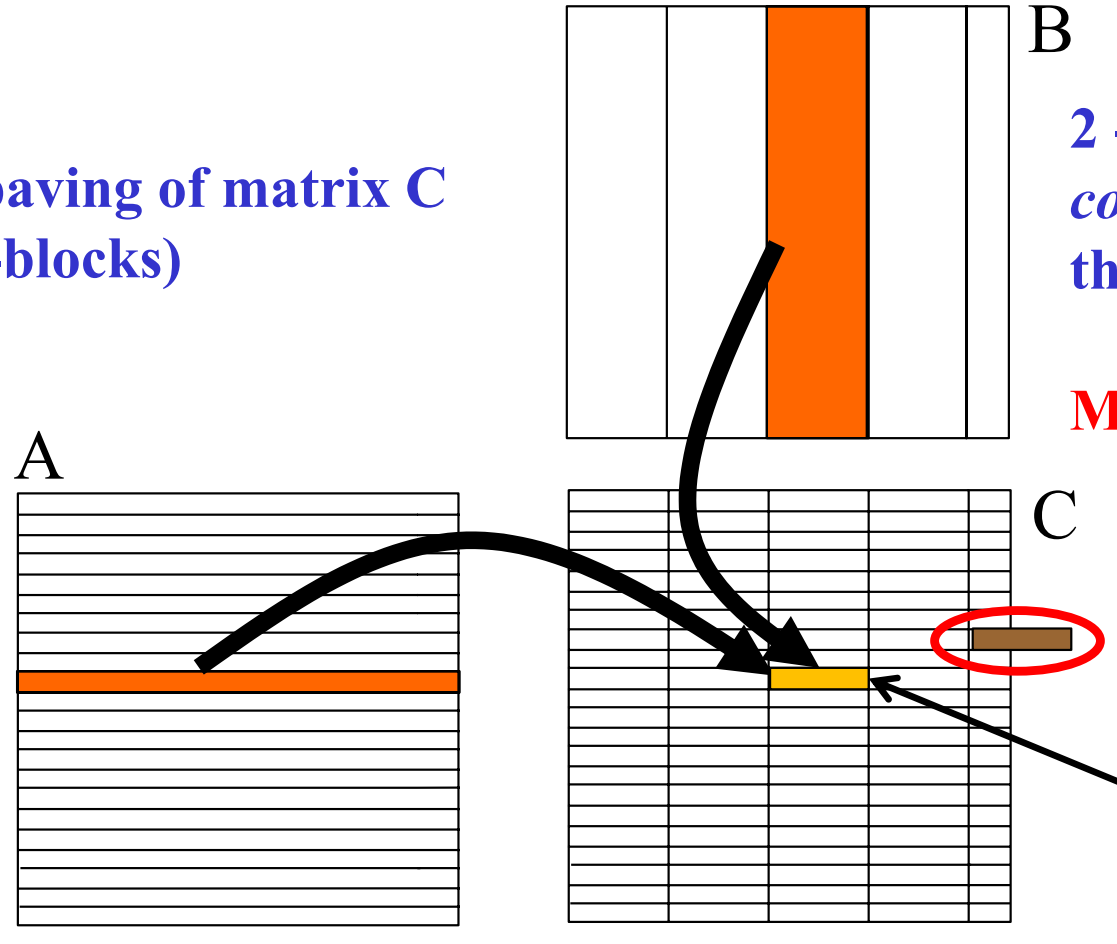
  Model : $T_{RAM\ access}^{total} = N_{RAM\ accesses}^{achieved\ by\ all\ warps} . t_{1\ RAM\ access}$

Ex1: 2D-grid of 1D-blocks – Boundaries

Product of matrices of *n*×*n* elements

- 1D-blocks of B_x threads (upgraded kernel k0)
- With: $n \neq k.B_x$, $k \in \mathbb{N}$

1 - Design the paving of matrix C (2D-grid of 1D-blocks)



2 - Implement *coalescent* accesses to the global memory

&

Manage *boundaries*

POLYTECH PARIS-SACLAY

Ex1: 2D-grid of 1D-blocks – Boundaries

Q1.5 - Upgrade of the 2D-grid of 1D-blocks

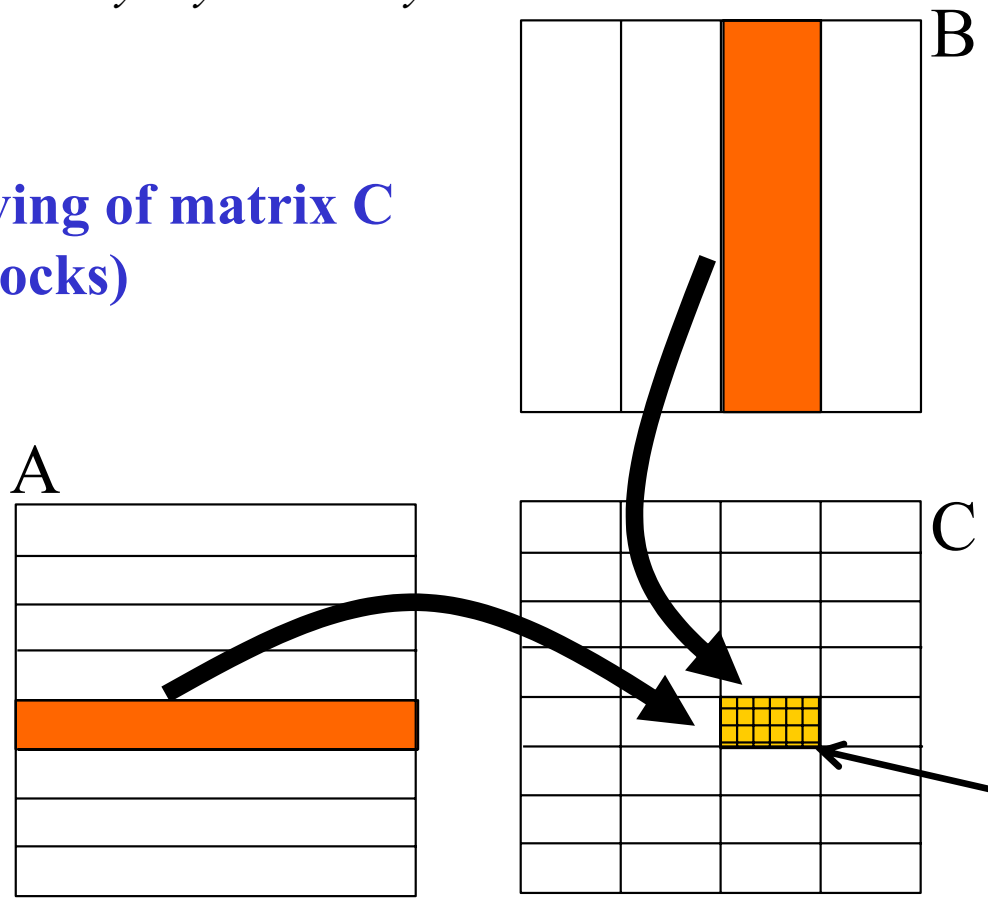
Q1.6 - Upgrade of the computing kernel

Ex2: 2D-grid of 2D-blocks

Product of matrices of *n*×*n* elements

- 2D-blocks of $B_x \times B_y$ threads (kernel k1)
- With: $n = k_x.B_x = k_y.B_y$, $(k_x, k_y) \in \mathbb{N}^2$

1 - Design the paving of matrix C (2D-grid of 2D-blocks)



2 - Implement *coalescent* accesses to the global memory

POLYTECH PARIS-SACLAY

Ex2: 2D-grid of 2D-blocks

Q2.1 - Definition of the 2D-grid of 2D-blocks

Q2.2 - Definition of the computing kernel

Q2.3 - Analysis of the coalescence

Q2.4 - Number of global memory accesses (not including cache memory):

- Compute the total number of memory accesses <u>requested</u> by the threads

$$N_{RAM\ accesses}^{requested\ by\ all\ threads} = (n_{threads} . n_{RAM\ accesses}^{requested\ by\ 1\ thread})$$

- Compute the total number of memory accesses <u>achieved</u> by the warps

  When accesses are coalescent: 1 warp accesses 32 data in $t_{1\ RAM\ access}$

  Model : $T_{RAM\ access}^{total} = N_{RAM\ accesses}^{achieved\ by\ all\ warps} . t_{1\ RAM\ access}$

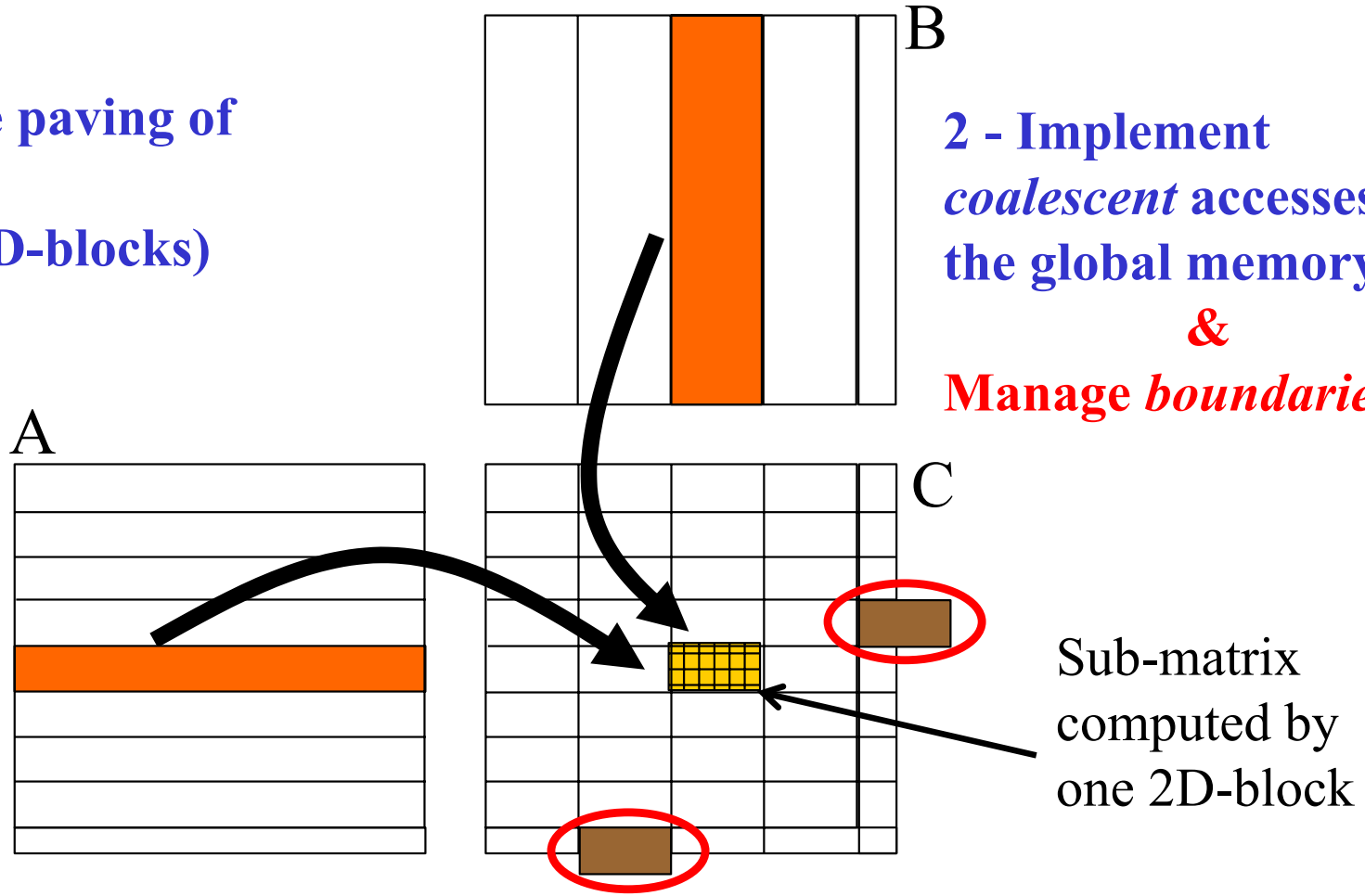
Ex2: 2D-grid of 2D-blocks – Boundaries

Product of matrices of *n*×*n* elements

- 2D-blocks of $B_x \times B_y$ threads (upgraded kernel k1)
- With: $n \neq k_x.B_x$, $n \neq k_y.B_y$, $(k_x, k_y) \in \mathbb{N}^2$

1 - Design the paving of matrix C (2D-grid of 2D-blocks)

2 - Implement *coalescent* accesses to the global memory & Manage *boundaries*



POLYTECH PARIS-SACLAY

Ex2: 2D-grid of 2D-blocks – Boundaries

Q2.5 - Upgrade of the 2D-grid of 2D-blocks

Q2.6 - Upgrade of the computing kernel

POLYTECH
PARIS-SACLAY

TD1 : Dense matrix product on GPU using registers

Fin