

# Modèles et techniques en programmation parallèle hybride et multi-cœurs

## Introduction au parallélisme multithreads

Marc Tajchman

CEA - DEN/DM2S/STMF/LMES

10/08/2020

# Parallélisme multi-threads en mémoire partagée

Exemple: si  $u$  et  $v$  sont des vecteurs de taille  $n > 4$ , on veut calculer

$$\begin{aligned}v[2] &= (u[1] + 2*u[2] + u[3]) / 4 \\v[3] &= (u[2] + 2*u[3] + u[4]) / 4\end{aligned}$$

Ces 2 instructions utilisent des composantes de  $u$  dont certaines sont communes ( $u[2]$  et  $u[3]$ ) et d'autres sont utilisées par une seule des instructions ( $u[1]$  et  $u[4]$ ).

Les composantes de  $u$  ne sont pas modifiées par les instructions. Les composantes de  $v$  sont modifiées (mais chaque instruction calcule une composante différente).

On remarque que le calcul de  $v[2]$  est indépendant de celui de  $v[3]$  et donc ces 2 calculs peuvent se faire en même temps par 2 exécutions différentes qui utilisent les mêmes vecteurs  $u$  et  $v$ .

1. Avant d'exécuter les 2 instructions:



Cœur 1



Cœur 2



Mémoire  
cache 1



Mémoire  
cache 2

Mémoire

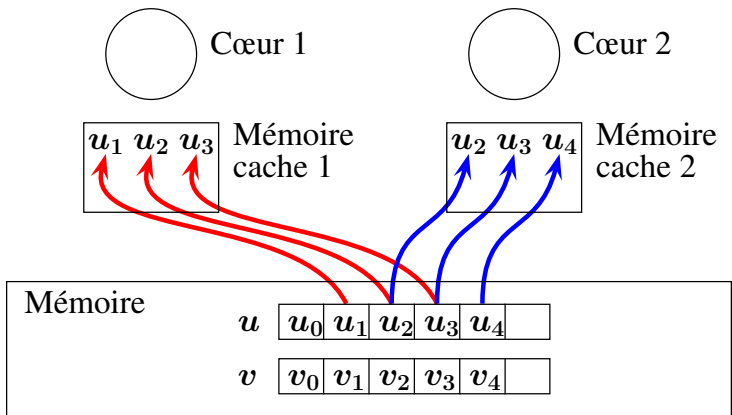
$u$ 

$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	
-------	-------	-------	-------	-------	--

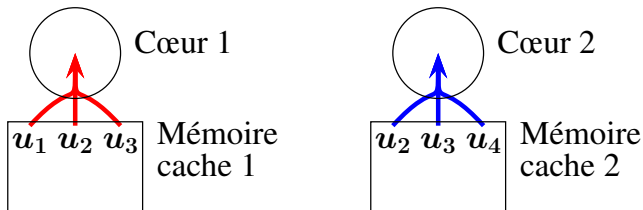
$v$ 

$v_0$	$v_1$	$v_2$	$v_3$	$v_4$	
-------	-------	-------	-------	-------	--

2. Les composantes de  $u$  sont copiées dans les mémoires cache:



3. Les composantes de  $u$  sont copiées dans les mémoires internes des processeurs:



Mémoire

$u$ 

$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	
-------	-------	-------	-------	-------	--

$v$ 

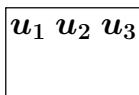
$v_0$	$v_1$	$v_2$	$v_3$	$v_4$	
-------	-------	-------	-------	-------	--

4. Le calcul est effectués dans les cœurs:

$$v_2 = (u_1 + 2u_2 + u_3)/4$$



Cœur 1

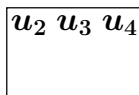


Mémoire  
cache 1

$$v_3 = (u_2 + 2u_3 + u_4)/4$$



Cœur 2



Mémoire  
cache 2

Mémoire

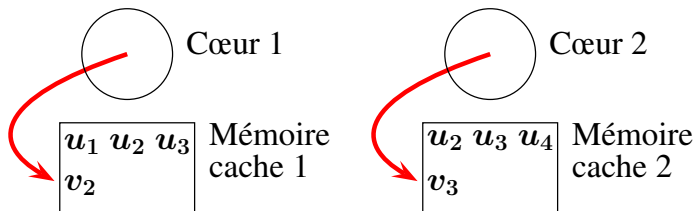
$u$ 

$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	
-------	-------	-------	-------	-------	--

$v$ 

$v_0$	$v_1$	$v_2$	$v_3$	$v_4$	
-------	-------	-------	-------	-------	--

5. Le résultat est recopié dans la mémoire cache:



Mémoire

$u$ 

$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	
-------	-------	-------	-------	-------	--

$v$ 

$v_0$	$v_1$	$v_2$	$v_3$	$v_4$	
-------	-------	-------	-------	-------	--

6. Le résultat est recopié dans la mémoire principale:

