

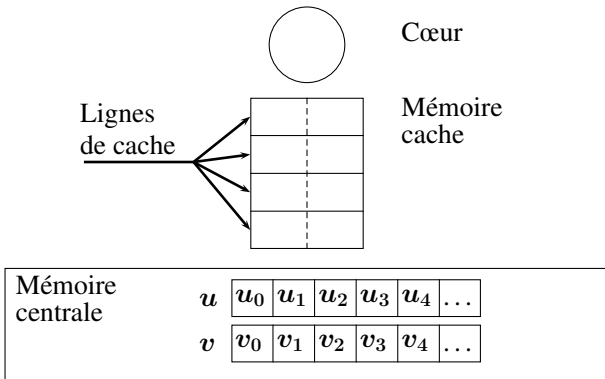
Une bonne utilisation de la mémoire est très importante pour l'optimisation de code.

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

$$\begin{aligned} v_0 &= u_0 \\ v_n &= u_n \\ \text{for}(i = 1; i < n - 1; i++) \\ &\quad v_i = (u_{i-1} + 2 * u_i + u_{i+1})/4; \end{aligned} \tag{1}$$

Supposons un système **idéalisé** par:

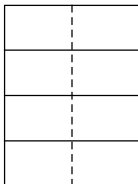
- un processeur (qui contient un seul cœur),
- un seul niveau de mémoire cache de taille 8 nombres réels (réparti en 4 lignes de cache de taille 2 nombres réels),
- la mémoire centrale



1. Avant d'exécuter $v_1 = (u_0 + 2 * u_1 + u_2)/4$:



Cœur

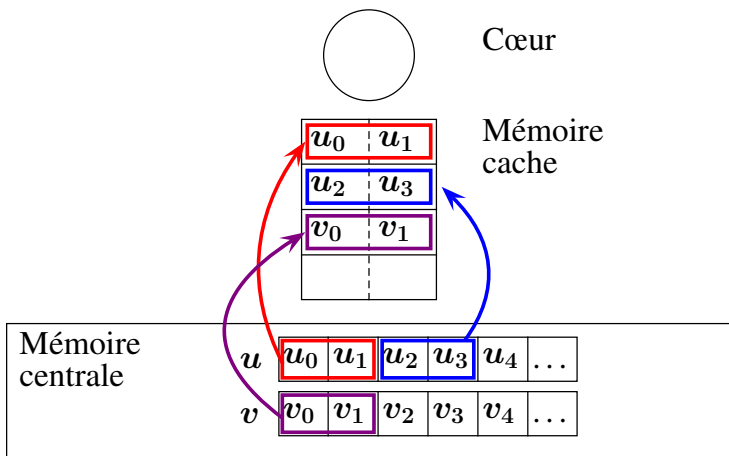


Mémoire
cache

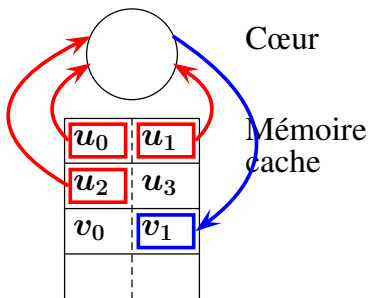
Mémoire
centrale

u	u_0	u_1	u_2	u_3	u_4	\dots
v	v_0	v_1	v_2	v_3	v_4	\dots

2. Les blocs contenant u_0 , u_1 , u_2 et v_1 (3 blocs) sont copiés dans la mémoire cache :



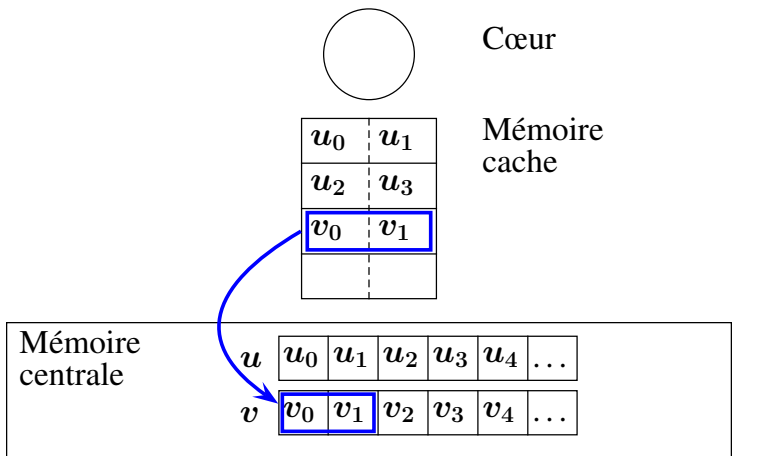
3. Le cœur utilise les copies de u_0 , u_1 , u_2 de la mémoire cache, calcule l'expression et place le résultat dans la mémoire cache :



Mémoire
centrale

u	u_0	u_1	u_2	u_3	u_4	...
v	v_0	v_1	v_2	v_3	v_4	...

4. Le bloc contenant le résultat est recopié dans la mémoire centrale :



5. Le calcul de l'instruction suivante $v_2 = (u_1 + 2 * u_2 + u_3) / 4$ peut commencer:



Cœur

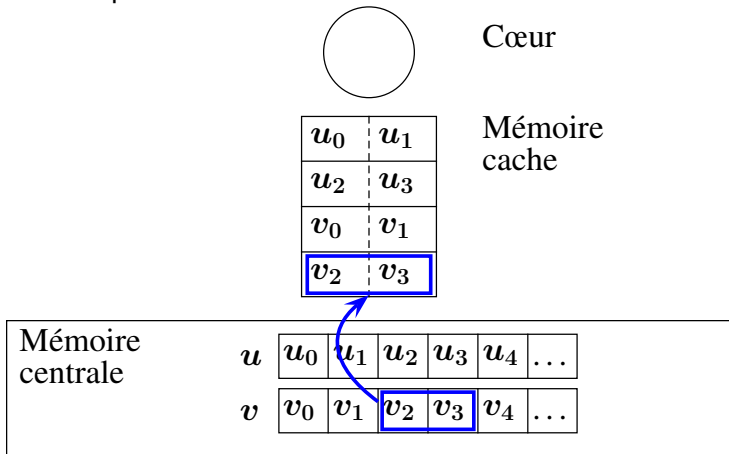
u_0	u_1
u_2	u_3
v_0	v_1

Mémoire
cache

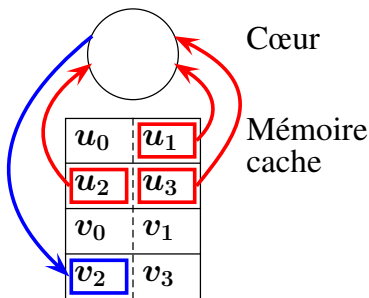
Mémoire
centrale

u	u_0	u_1	u_2	u_3	u_4	...
v	v_0	v_1	v_2	v_3	v_4	...

6. Les composantes de u nécessaires **sont déjà dans la mémoire cache**, seul le bloc contenant la composante v_2 doit être copié dans la mémoire cache :



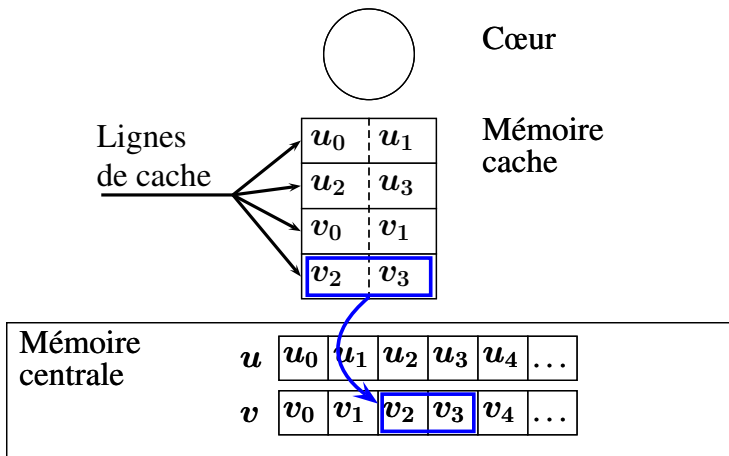
7. Le cœur utilise les copies de u_1 , u_2 , u_3 de la mémoire cache, calcule l'expression et place le résultat dans la mémoire cache :



Mémoire
centrale

u	u_0	u_1	u_2	u_3	u_4	...
v	v_0	v_1	v_2	v_3	v_4	...

8. Le bloc contenant le résultat est recopié dans la mémoire centrale :



En résumé :

- ▶ La première instruction $v_1 = (u_0 + 2 * u_1 + u_2)/4$ utilise
 - ▶ 4 transferts (lents) mémoire centrale - mémoire cache
 - ▶ 4 transferts (rapides) mémoire cache - cœur
- ▶ La deuxième instruction $v_2 = (u_1 + 2 * u_2 + u_3)/4$ utilise
 - ▶ 2 transferts (lents) mémoire centrale - mémoire cache
 - ▶ 4 transferts (rapides) mémoire cache - cœur

Attention : cet exemple est (très) simplifié : en général la mémoire cache est de taille plus grande que dans l'exemple et, de plus, il y a plusieurs types de mémoire cache dans un ordinateur

Dans les processeurs actuels, les lignes de cache ont une taille de quelques dizaines d'octets (entre 32 et 128 octets).