Comportement et exploitation de la cache en multiprogrammation

Félix-Antoine Ouellet

Université de Sherbrooke

20 novembre 2014

Plan

Un simple problème

Étape 1

Générer aléatoirement N entiers et les insérer dans une séquence de sorte qu'ils soient triés en ordre croissant.
Par exemple, 5 1 4 2 donne:

- 5
- 1 5
- 1 4 5
- 1 2 4 5

Un simple problème

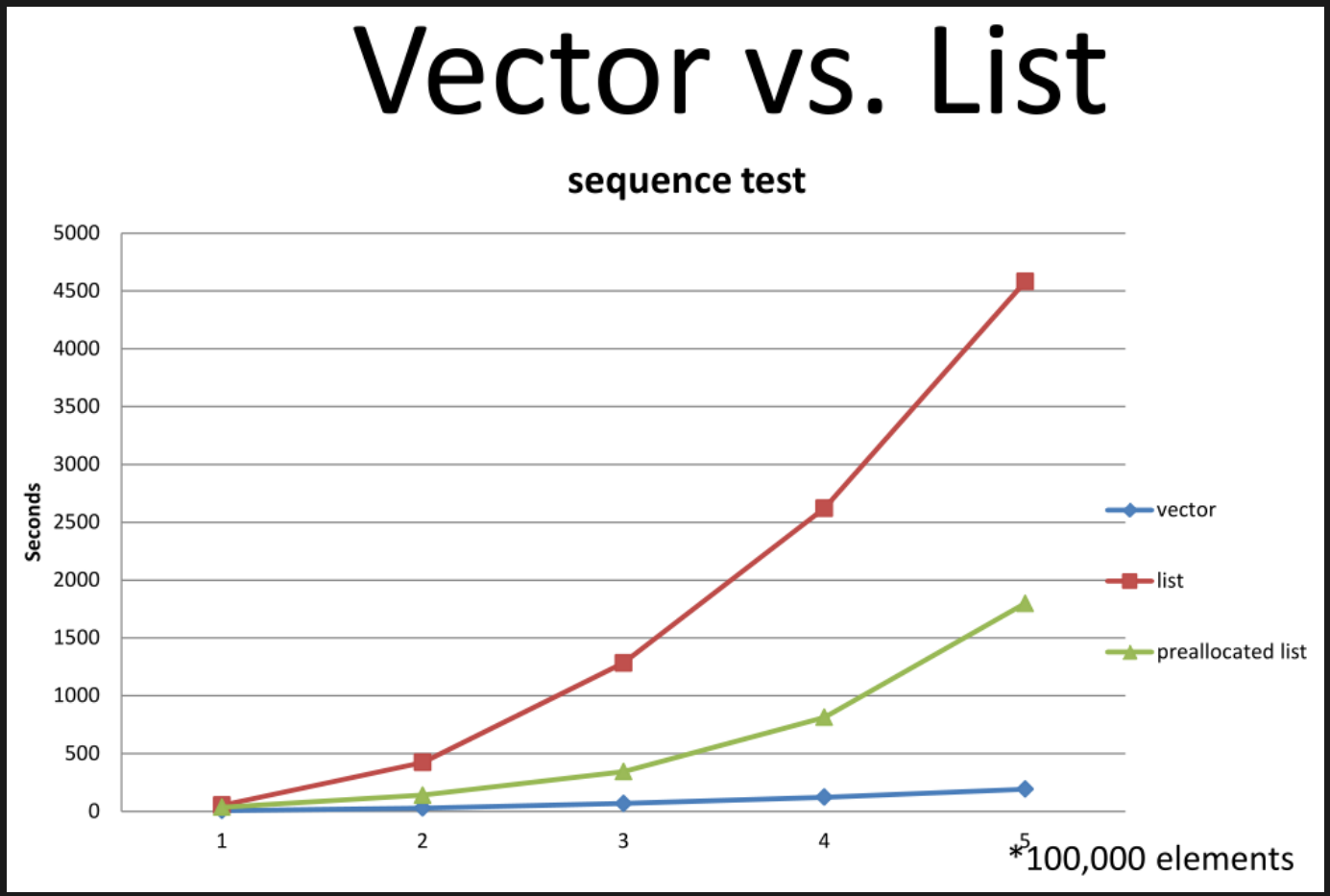
Étape 2

Enlever les éléments de la séquence 1 à 1, et ce, de manière aléatoire.
Par exemple, 1 2 0 0 donne:

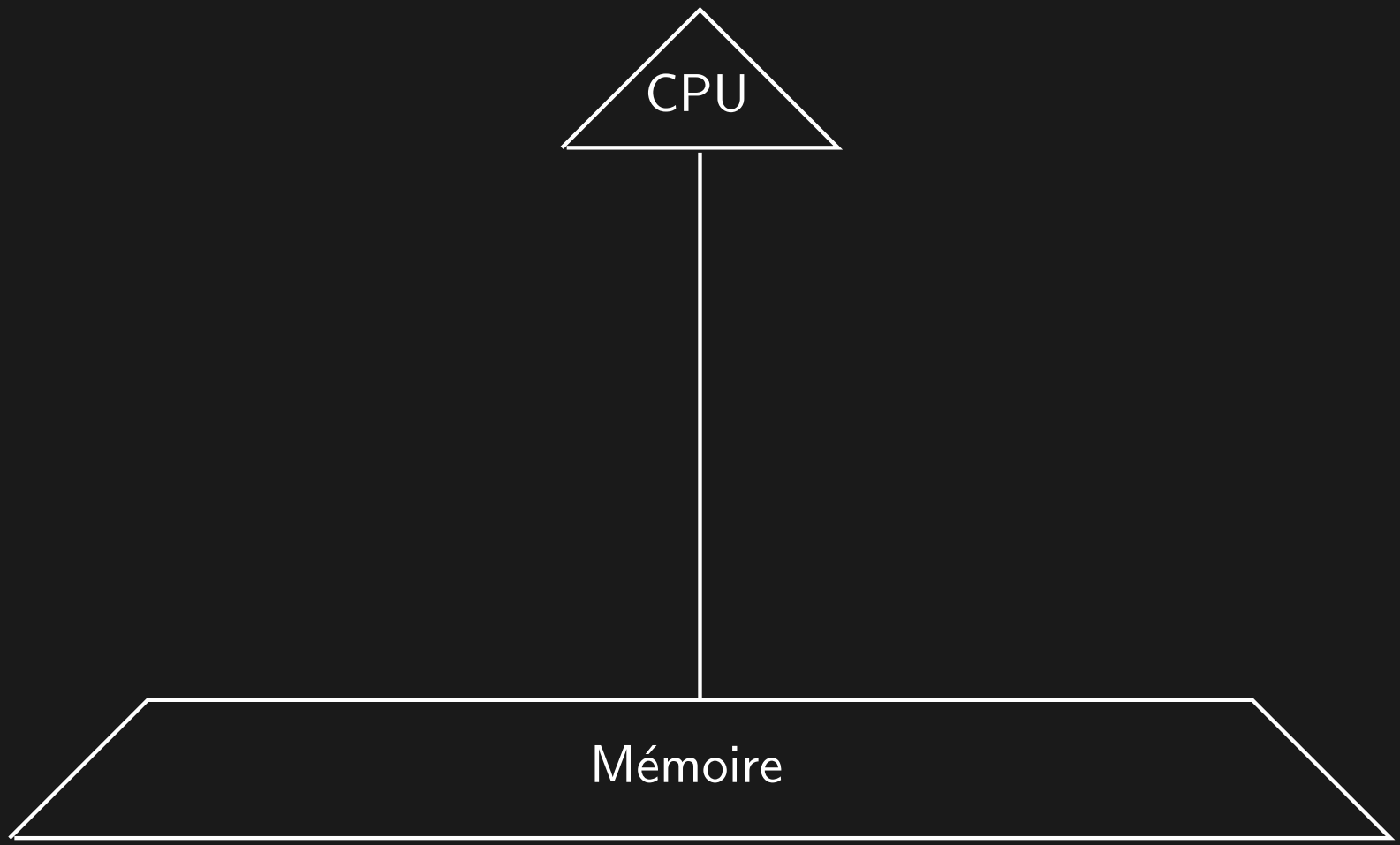
- 1 2 4 5
- 1 4 5
- 1 4
- 4

Un simple problème

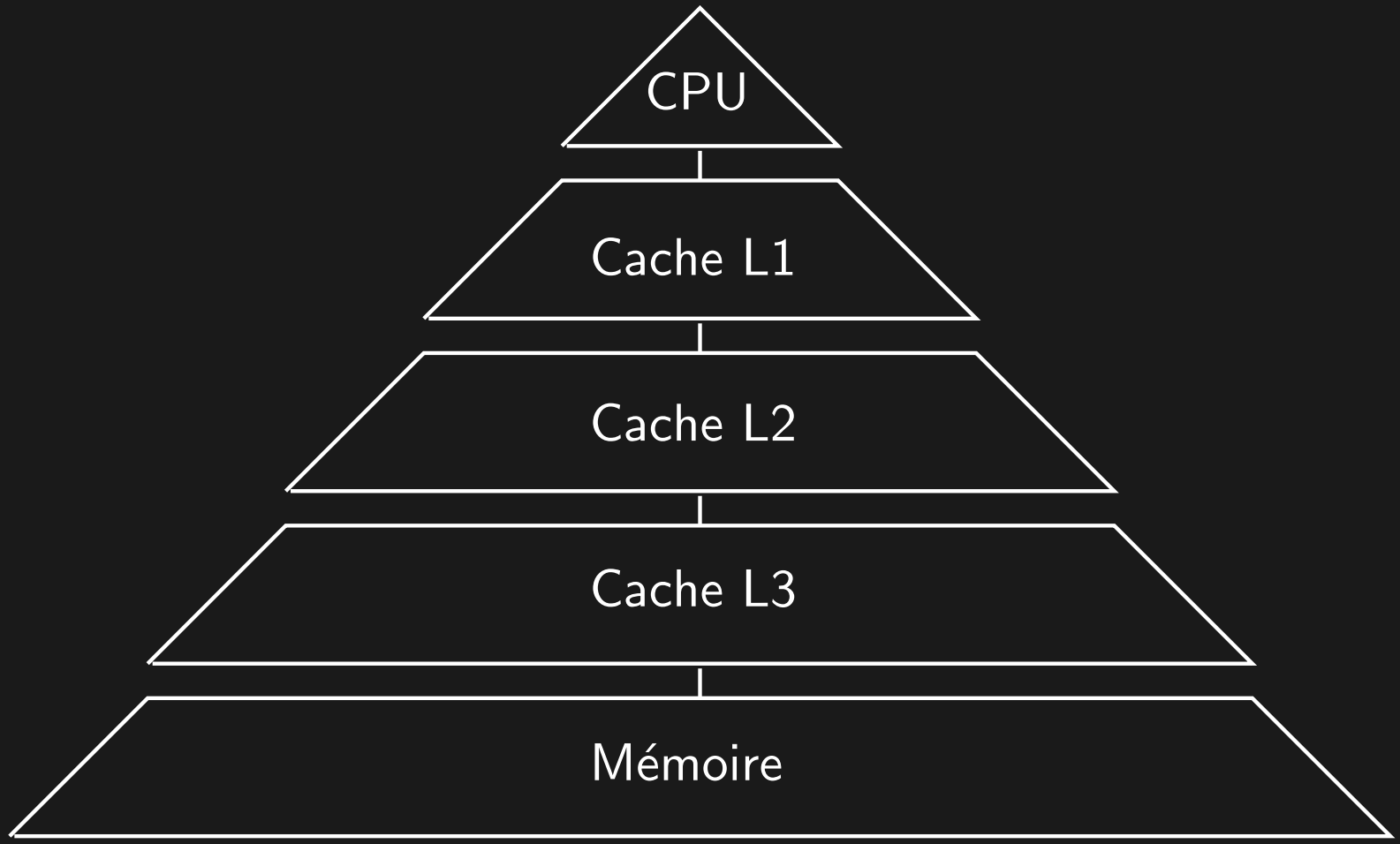
Résultats



Modèle de base



Réalité



Plan

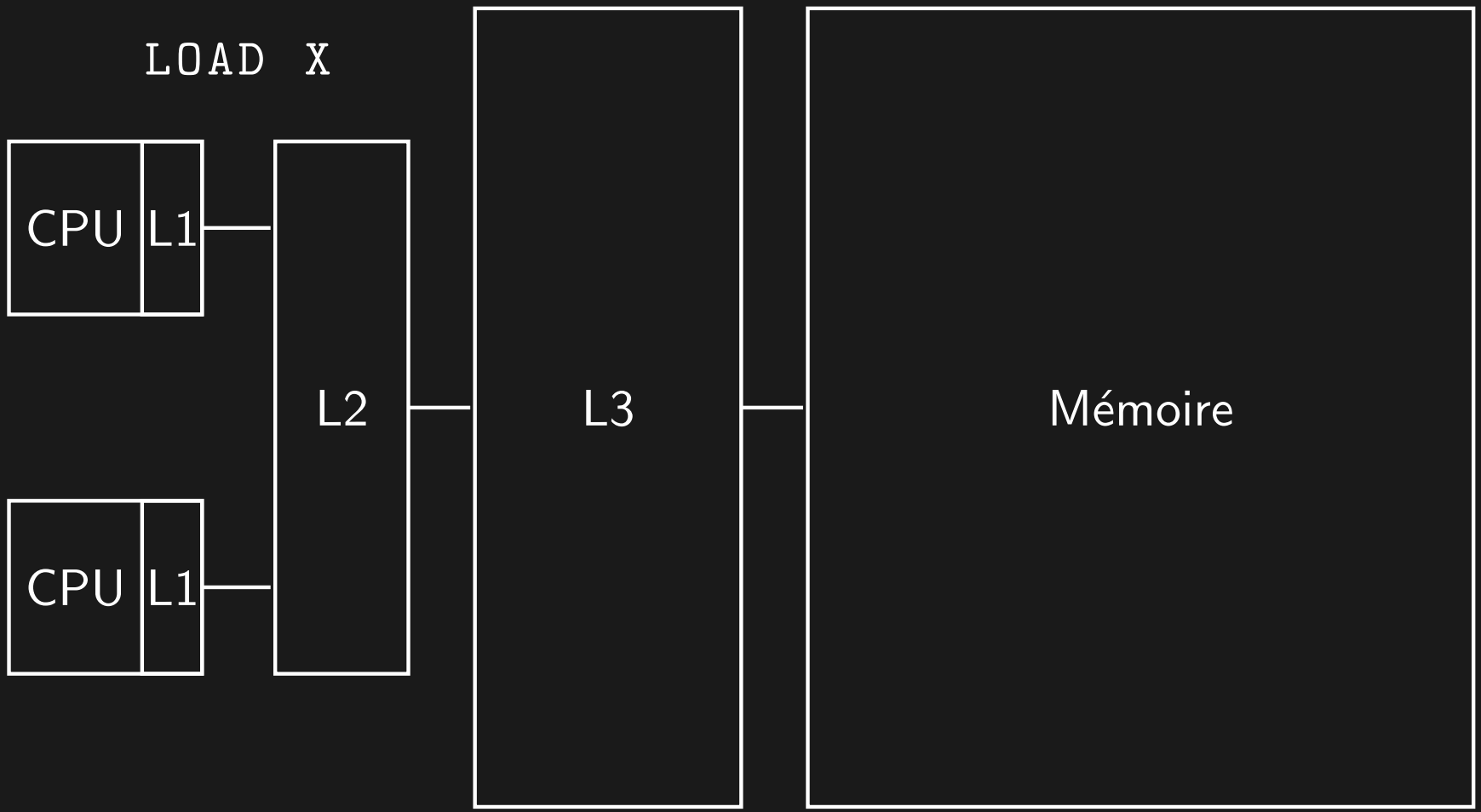
Cache
Illustration

- Comporte B blocs (*cache lines*)
- Taille M
- Chaque bloc à une taille B/M

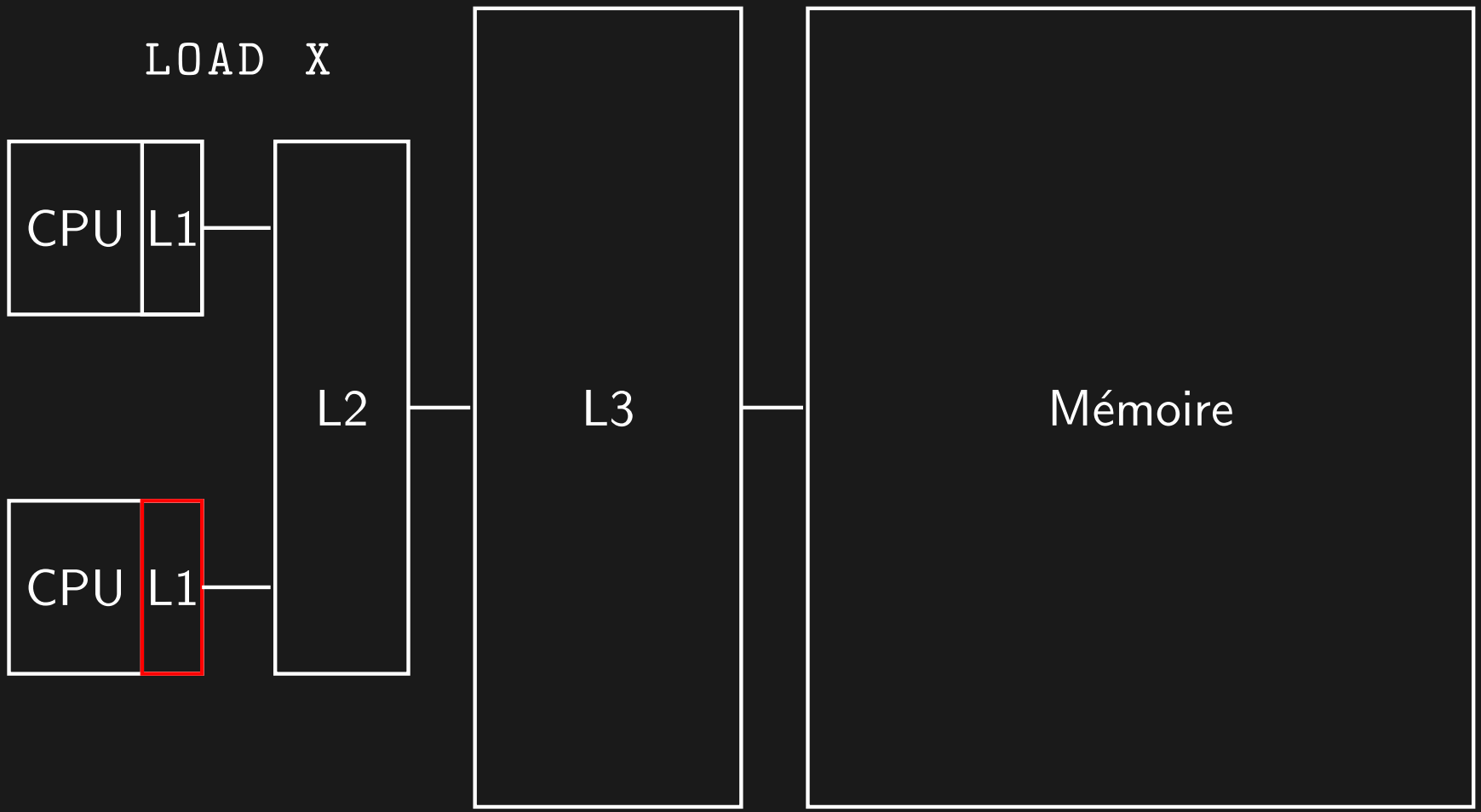
Accès à la mémoire

- En général, les processeurs ne peuvent accéder directement à la mémoire
- Les accès mémoire se font au travers d'une hiérarchie de caches

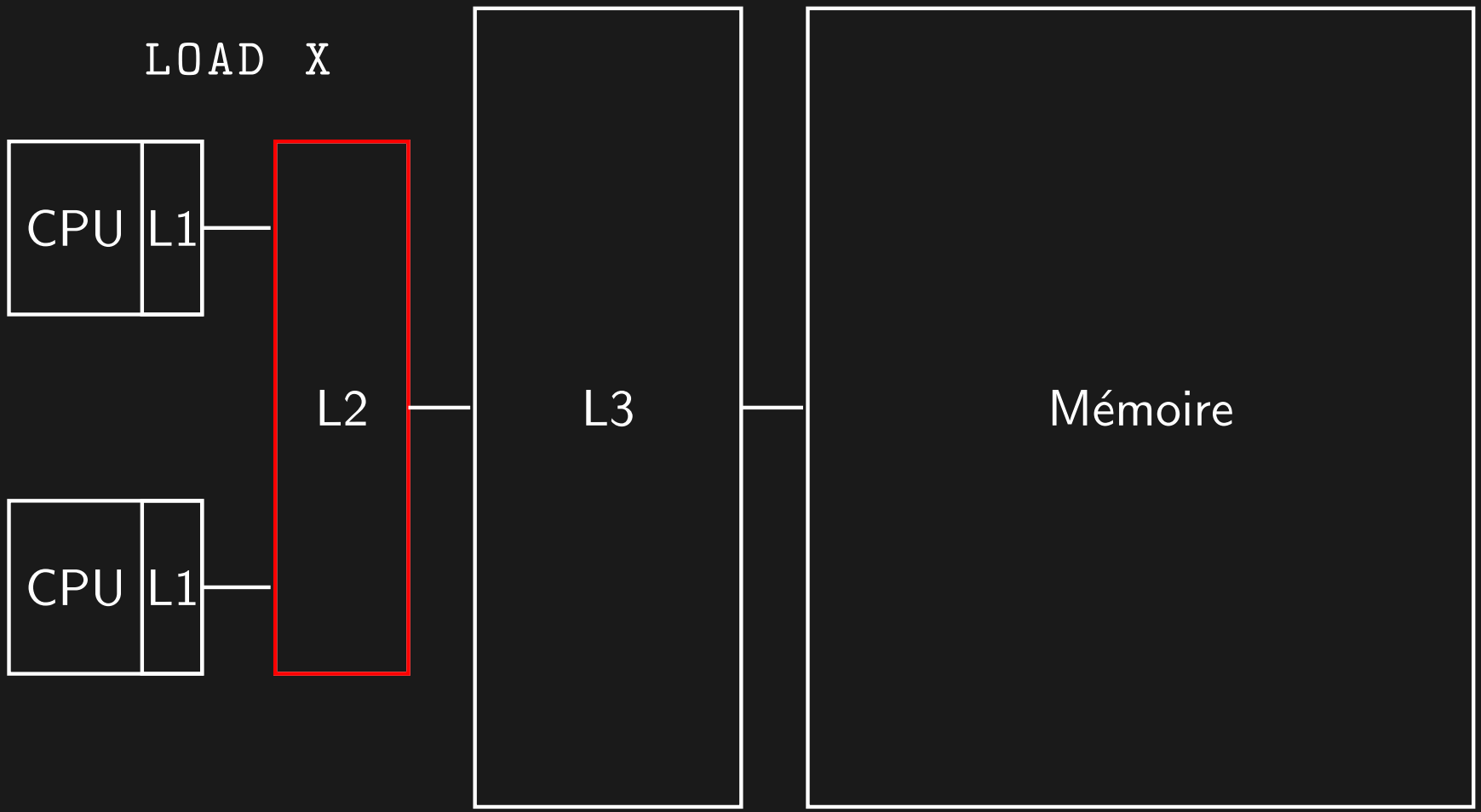
Lecture



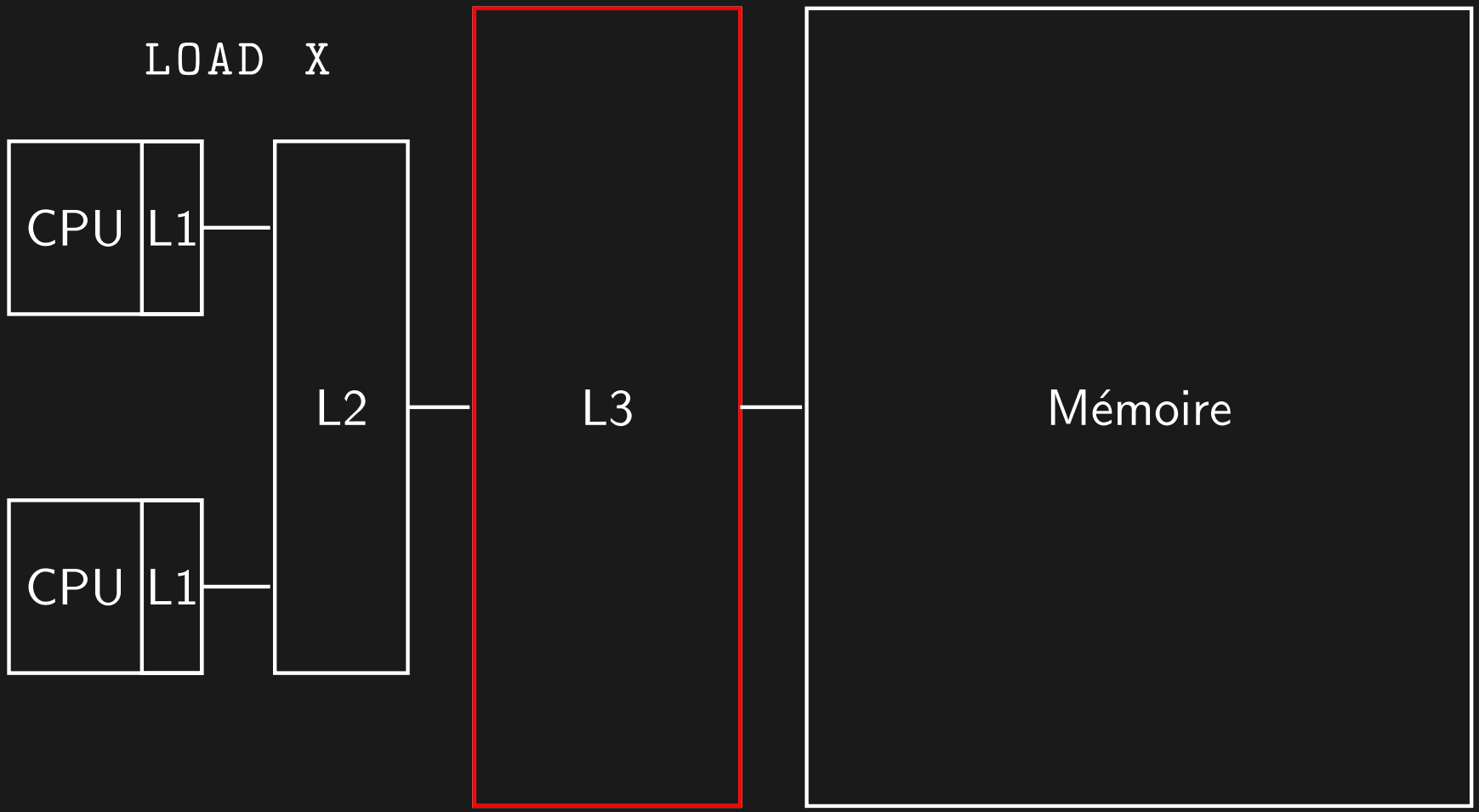
Lecture



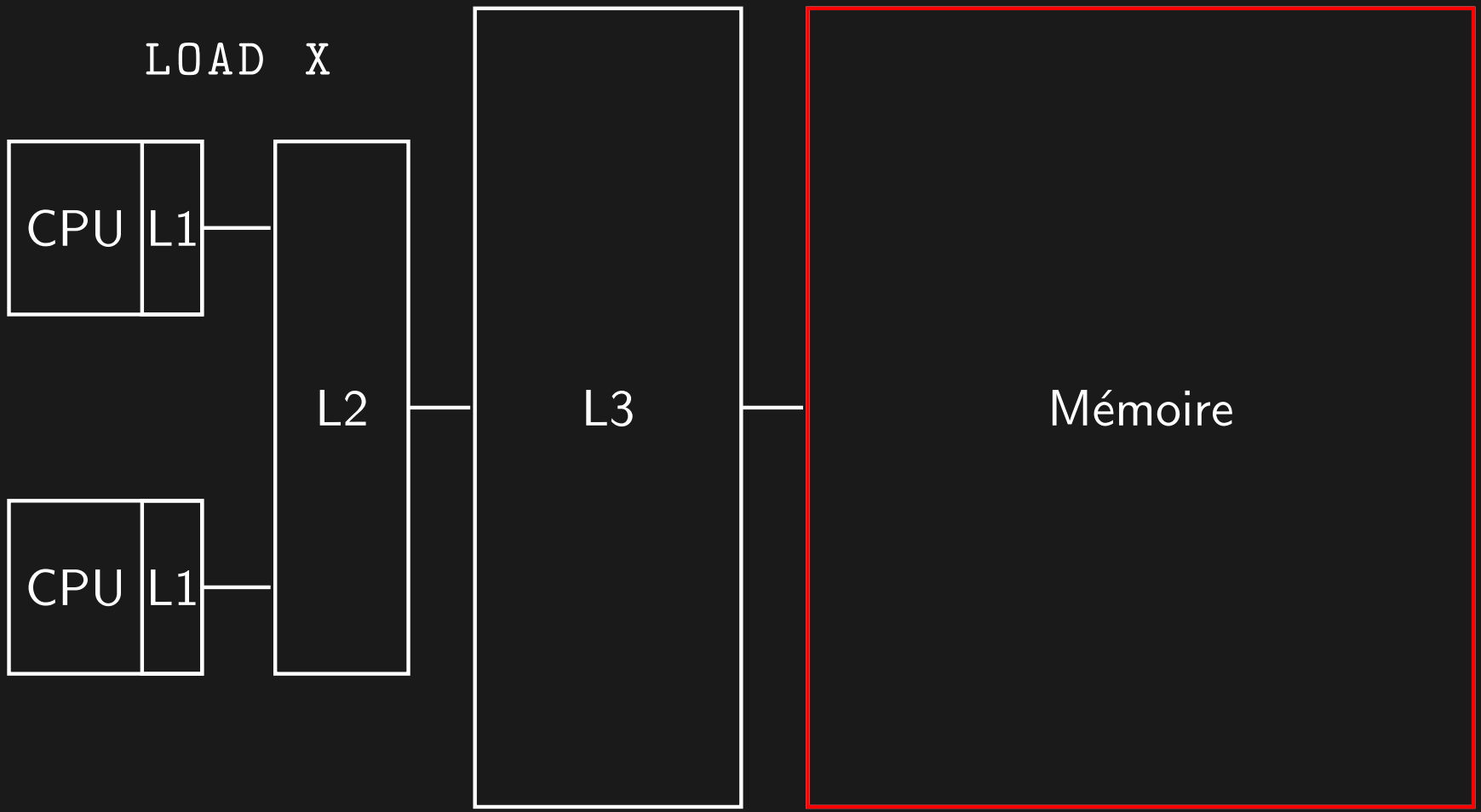
Lecture



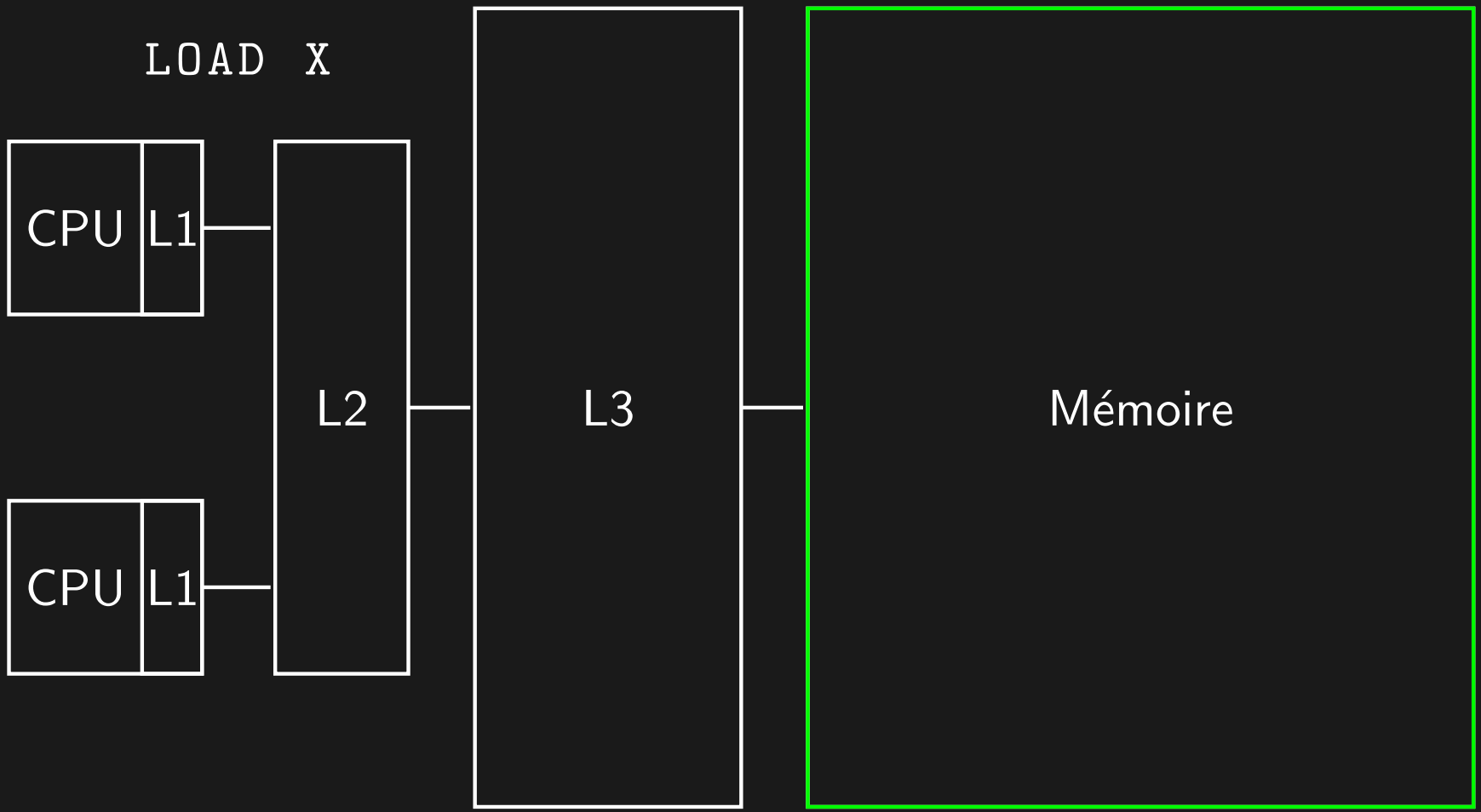
Lecture



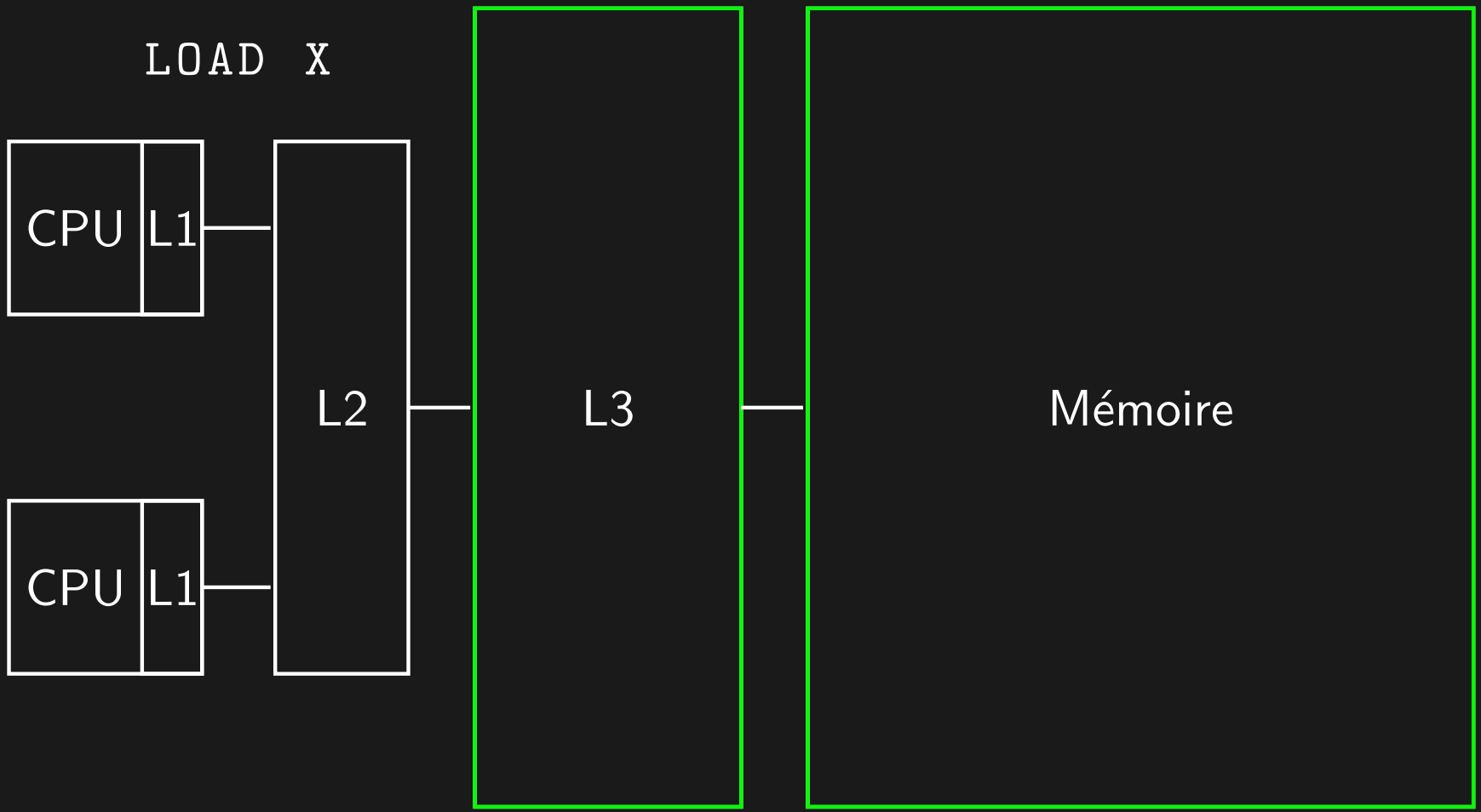
Lecture



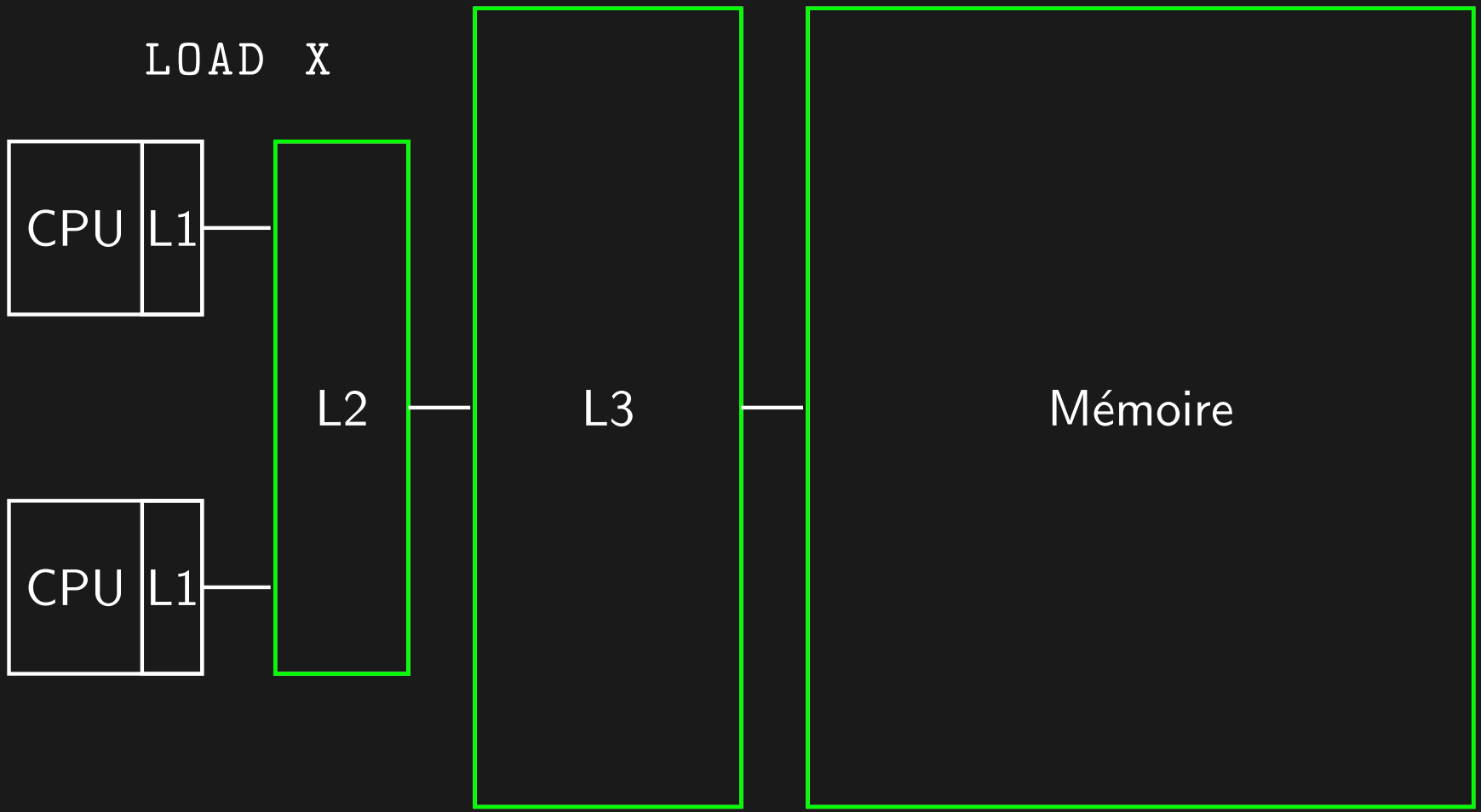
Lecture



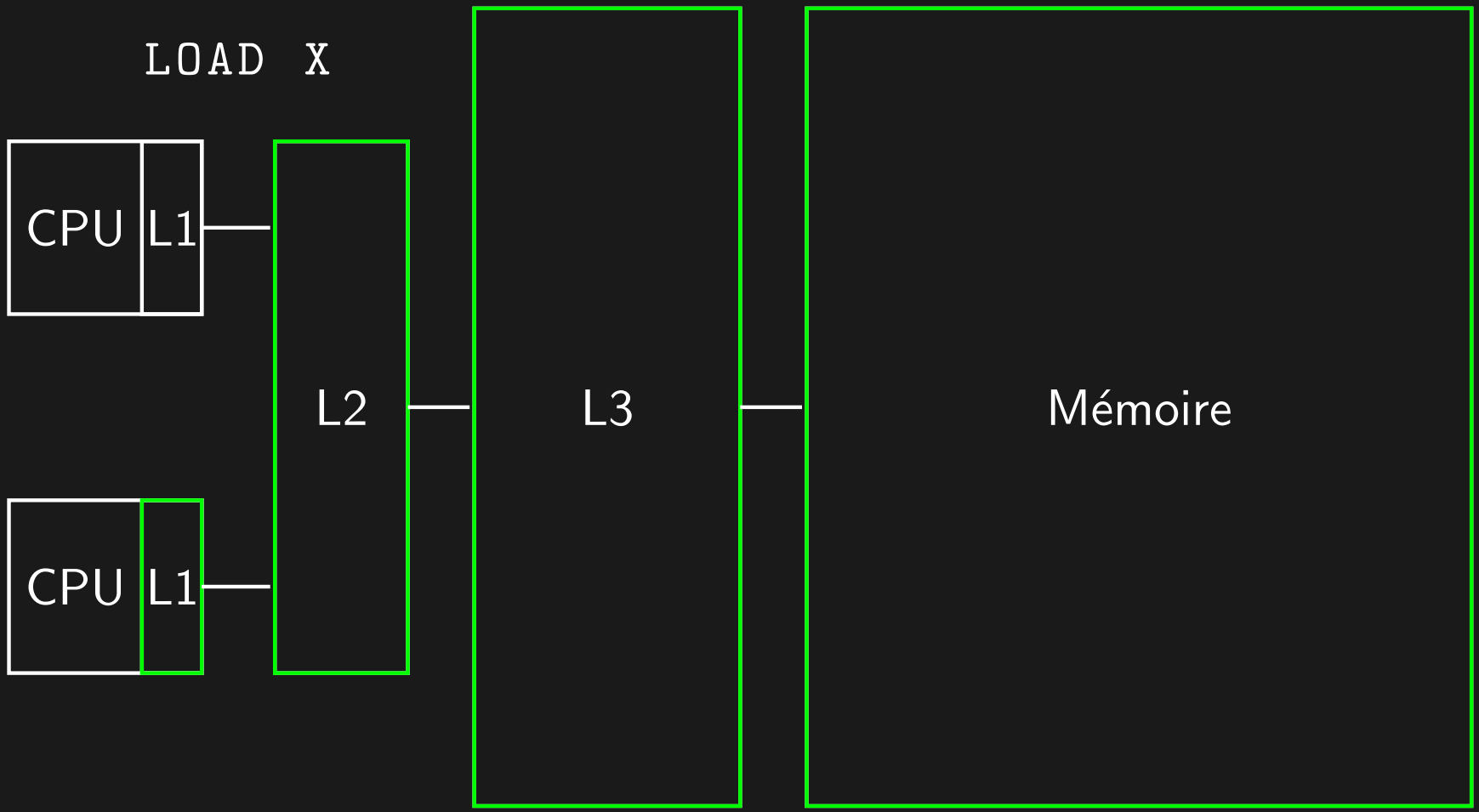
Lecture



Lecture



Lecture



Écriture

Write-through

- Le contenu de la cache et de la mémoire sont mis à jour simultanément
- Le contenu de chaque cache est identique au contenu de la mémoire en tout temps
- Minimise la perte de données

Écriture
Write-back

- La mise à jour de la mémoire est différée à des moments précis
  - Exemple: Lecture de l'emplacement mémoire, *cache line* sur le point de se faire évincer
- Offre des gains de performance en vitesse

Problème

Que se passe-t-il dans un contexte parallèle?

Protocole de cohérence

- Assure que le contenu de multiples caches demeurent cohérent
- *Directory-based* ou *snooping*
- *Write-through* facile à gérer
- *Write-back* plus subtile

Protocole MESI

Définitions

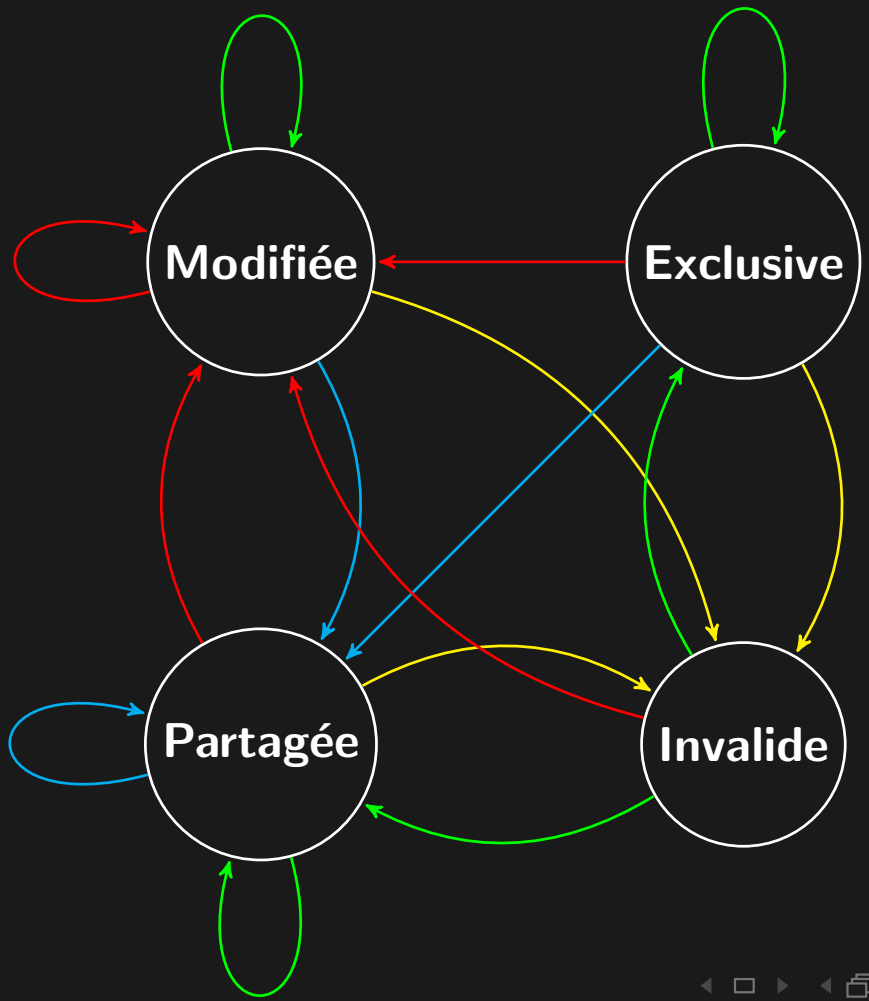
4 états possibles:

- M $\rightarrow$ Modifiée: Copie modifiée d'une valeur en mémoire
- E $\rightarrow$ Exclusive: Copie propre d'une donnée en mémoire présente uniquement dans la cache courante
- S $\rightarrow$ Partagée: Copie propre d'une donnée en mémoire que plusieurs caches peuvent posséder
- I $\rightarrow$ Invalide: Ne peut être utilisée

Protocole MESI

Illustration

Écriture locale
Lecture locale
Lecture distante
Écriture distante



Plan

Concept

Algorithmes

Tri rapide

Structures de données

Arbre de van Emde Boas

Plan

Conclusion

- Les meilleures performances proviennent souvent d'un programme conscient de la cache.

Conclusion

- Les meilleures performances proviennent souvent d'un programme conscient de la cache.
- Une bonne utilisation de la cache en programmation parallèle demande par contre plus de finesse