



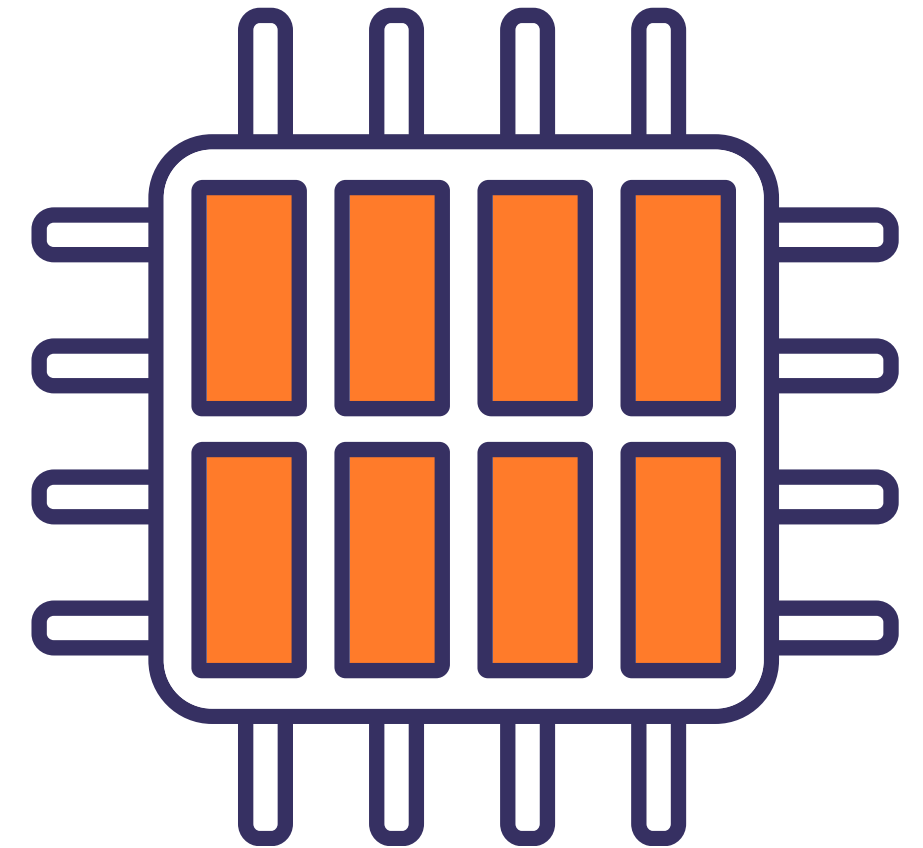
SD – Intro

Architecture mémoire

Programmation parallèle

Distributed systems

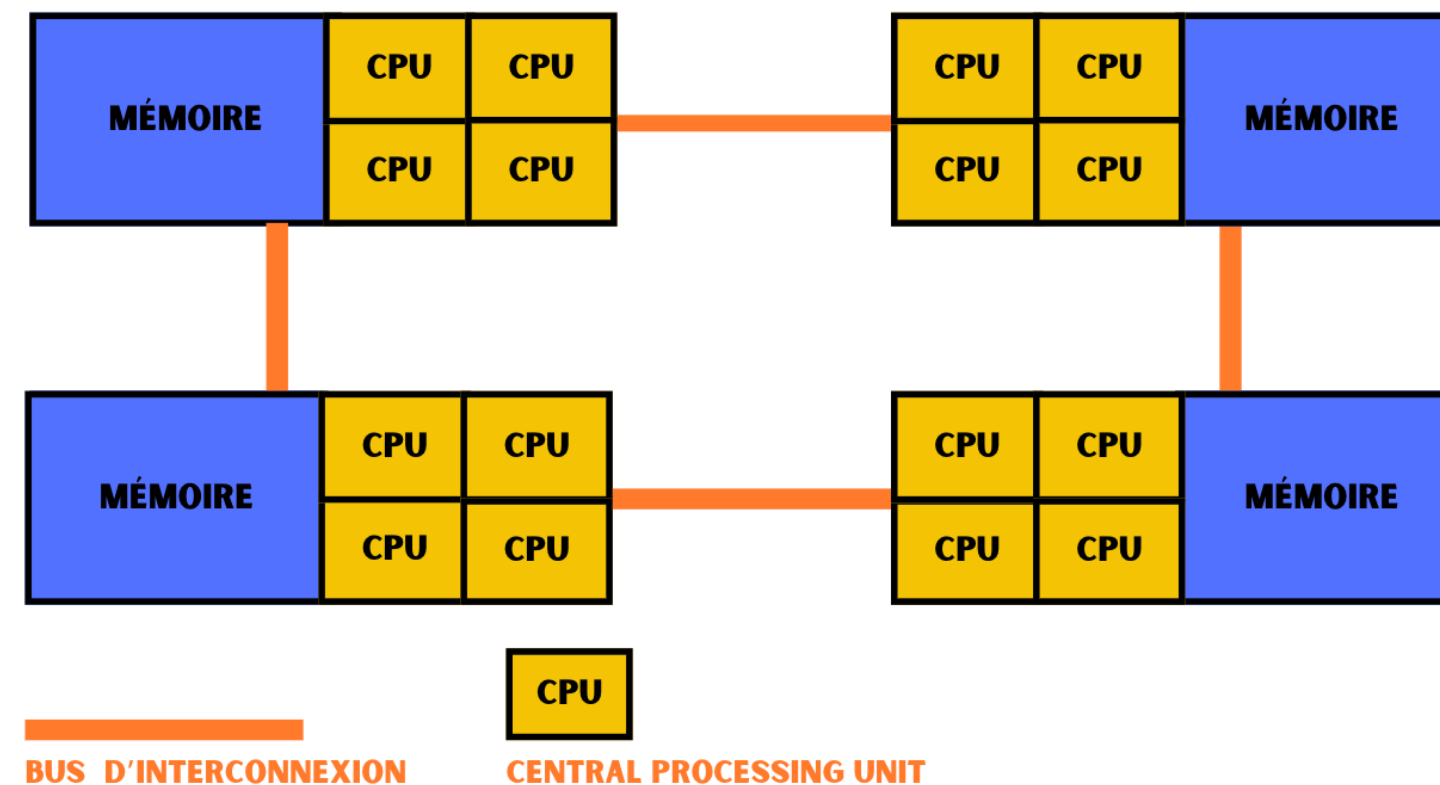
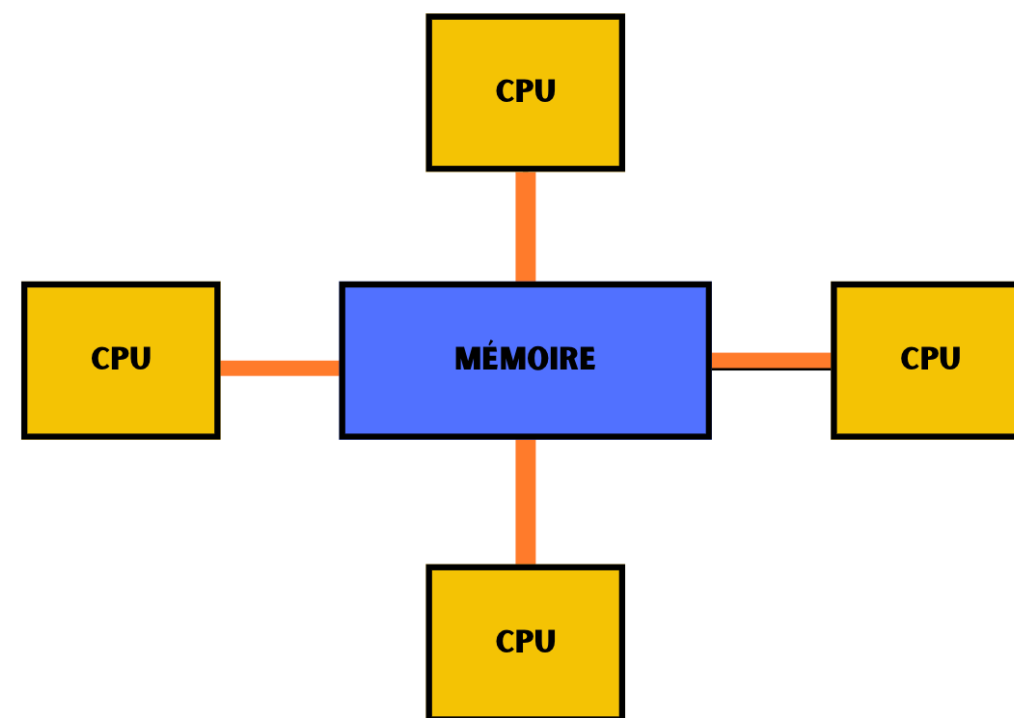
Last update: Fév. 2024



ORDINATEURS À MÉMOIRE PARTAGÉE

Historiquement, les machines à mémoire partagée ont été classées en **UMA** et **NUMA**, en fonction des temps d'accès à la mémoire.

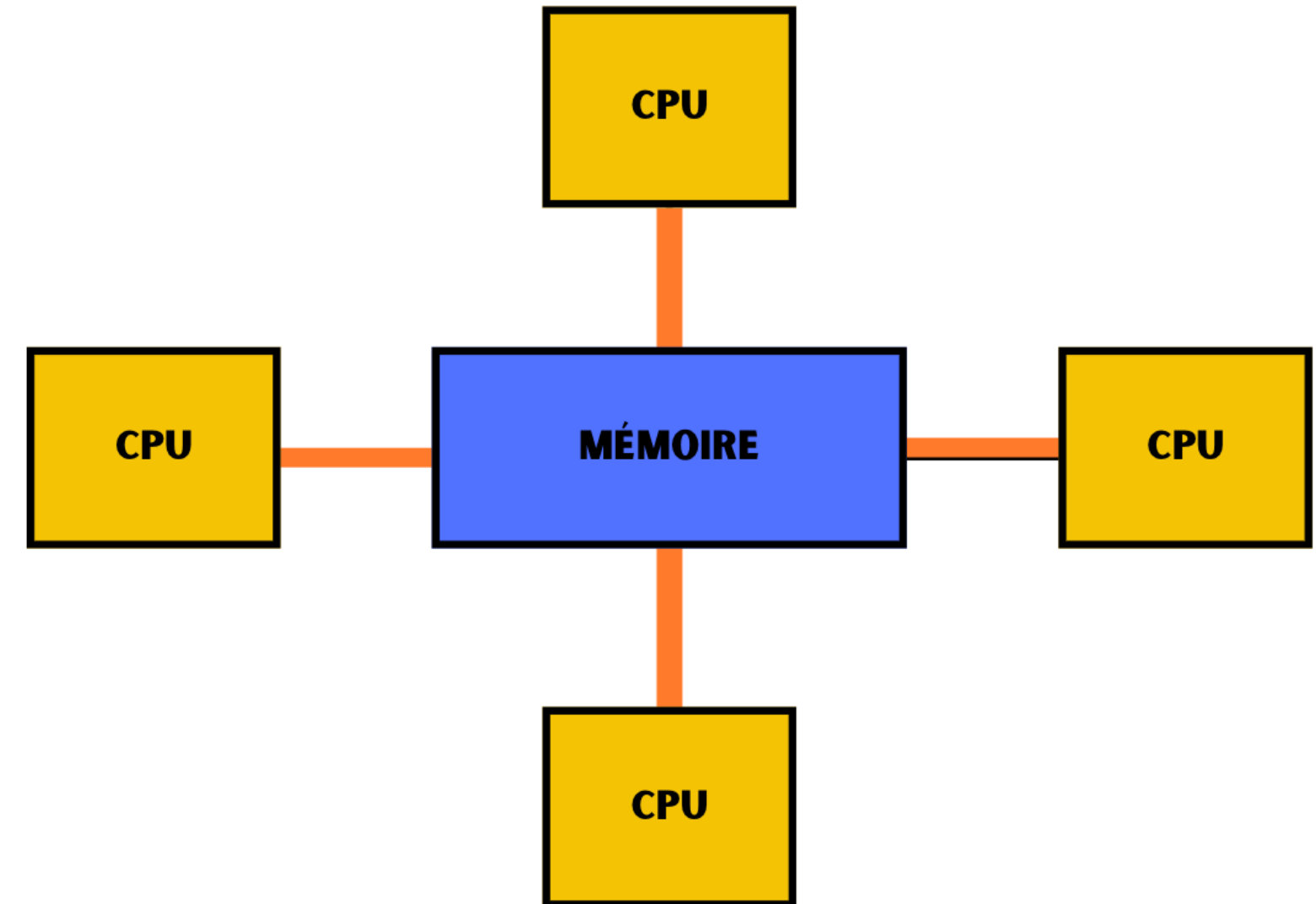
- **Accès Mémoire Uniforme (UMA)**
- **Accès Mémoire Non-Uniforme (NUMA)**



MÉMOIRE PARTAGÉE - ACCÈS UNIFORME (UMA)

Accès Mémoire Uniforme (UMA) :

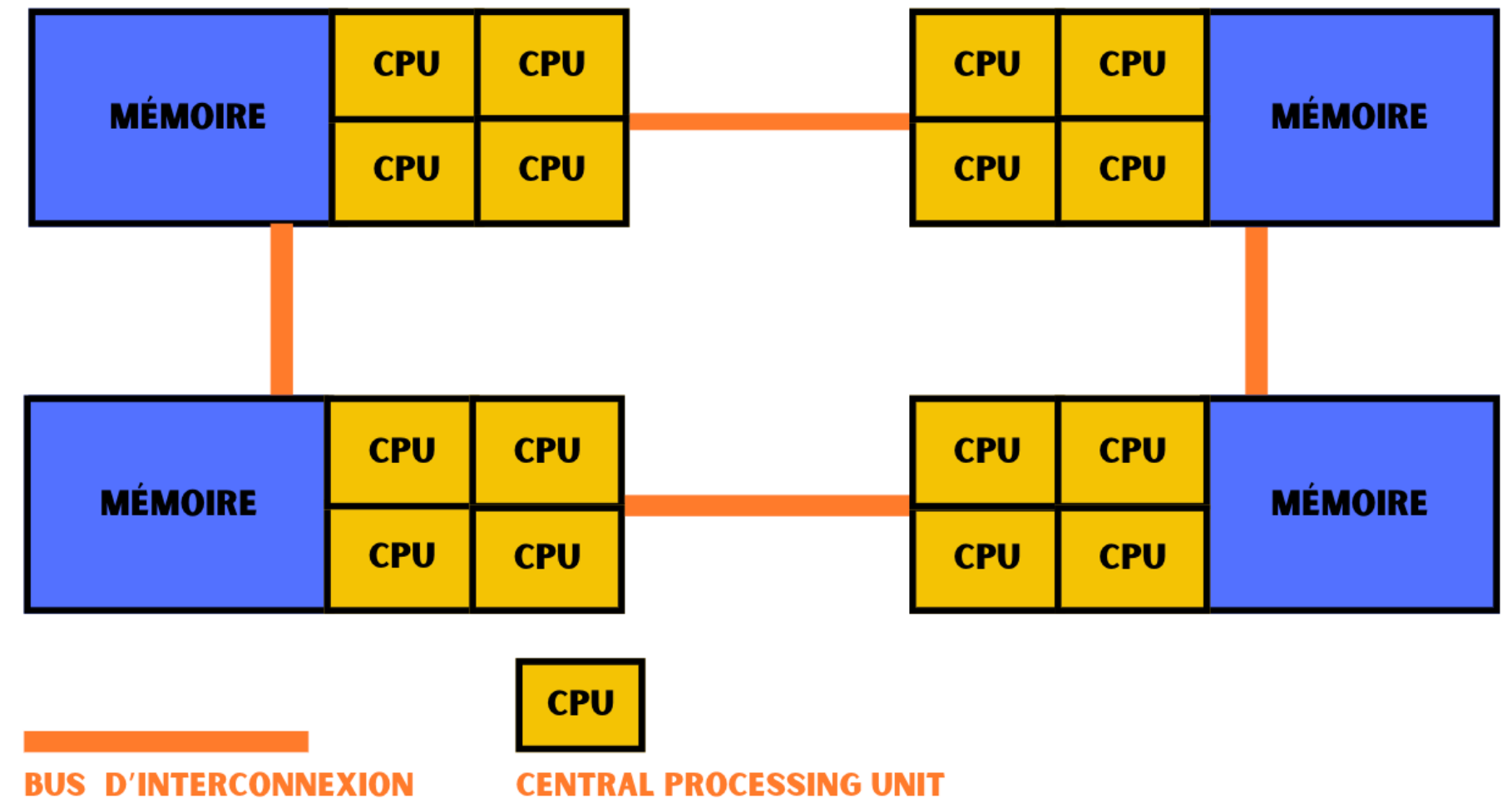
- Représenté le plus couramment aujourd'hui par les machines à processeurs symétriques (SMP).
- Ces machines ont des processeurs identiques avec un accès égal et des temps d'accès à la mémoire uniformes.
- Appelé aussi CC-UMA (Uniform Memory Access Cache Coherent), qui signifie que si un processeur met à jour un emplacement de mémoire partagée, tous les autres processeurs sont informés de la mise à jour.



MÉMOIRE PARTAGÉE - ACCÈS NON-UNIFORME

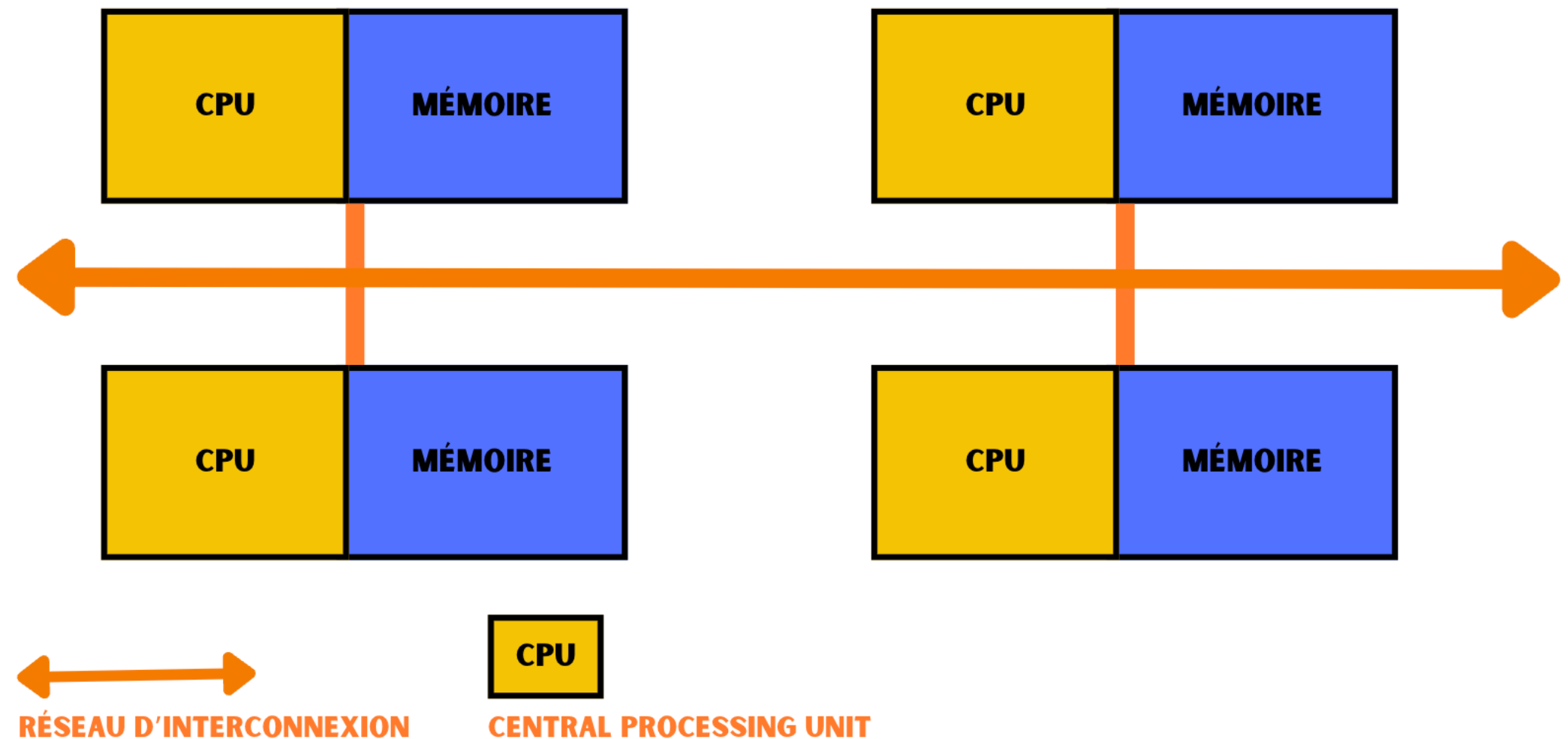
Accès Mémoire Non-Uniforme (NUMA) :

- Souvent réalisé en reliant physiquement deux ou plusieurs SMPs, où un SMP peut accéder directement à la mémoire d'un autre SMP.
- Tous les processeurs n'ont pas un temps d'accès égal à toutes les mémoires et l'accès à la mémoire à travers la liaison est plus lent.
- Si la cohérence de cache est maintenue, cela peut également être appelé CC-NUMA (Cache Coherent NUMA).



MÉMOIRE DISTRIBUÉE

- Les systèmes à mémoire distribuée nécessitent un réseau de communication pour connecter la mémoire entre les processeurs.
- Chaque processeur possède sa propre mémoire locale, sans concept d'espace d'adressage global entre processeurs.
- La responsabilité du programmeur inclut la définition explicite de la communication des données entre processeurs et la synchronisation des tâches, car chaque processeur fonctionne indépendamment.



MÉMOIRE PARTAGÉE - DISTRIBUTÉE HYBRIDE

Les plus grands et les plus rapides ordinateurs dans le monde aujourd'hui utilisent à la fois des architectures de mémoire partagée et distribuée.

