



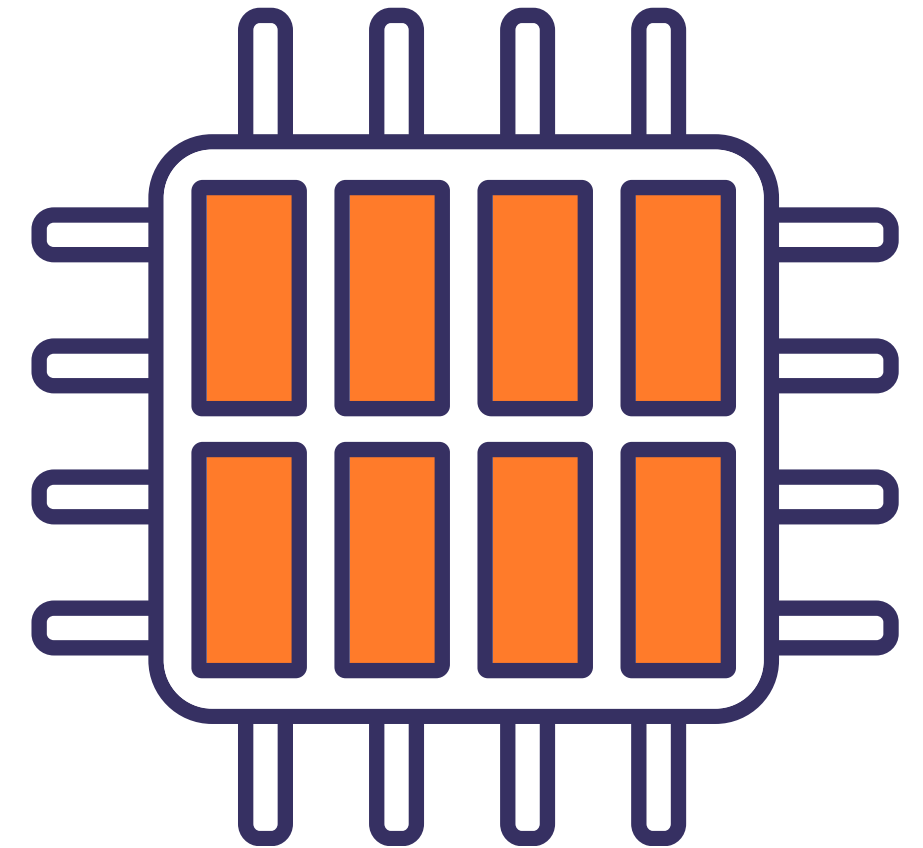
SD – intro

Synchronization, Grains et I/O

Programmation parallèle

Distributed systems

Last update: Fév. 2024

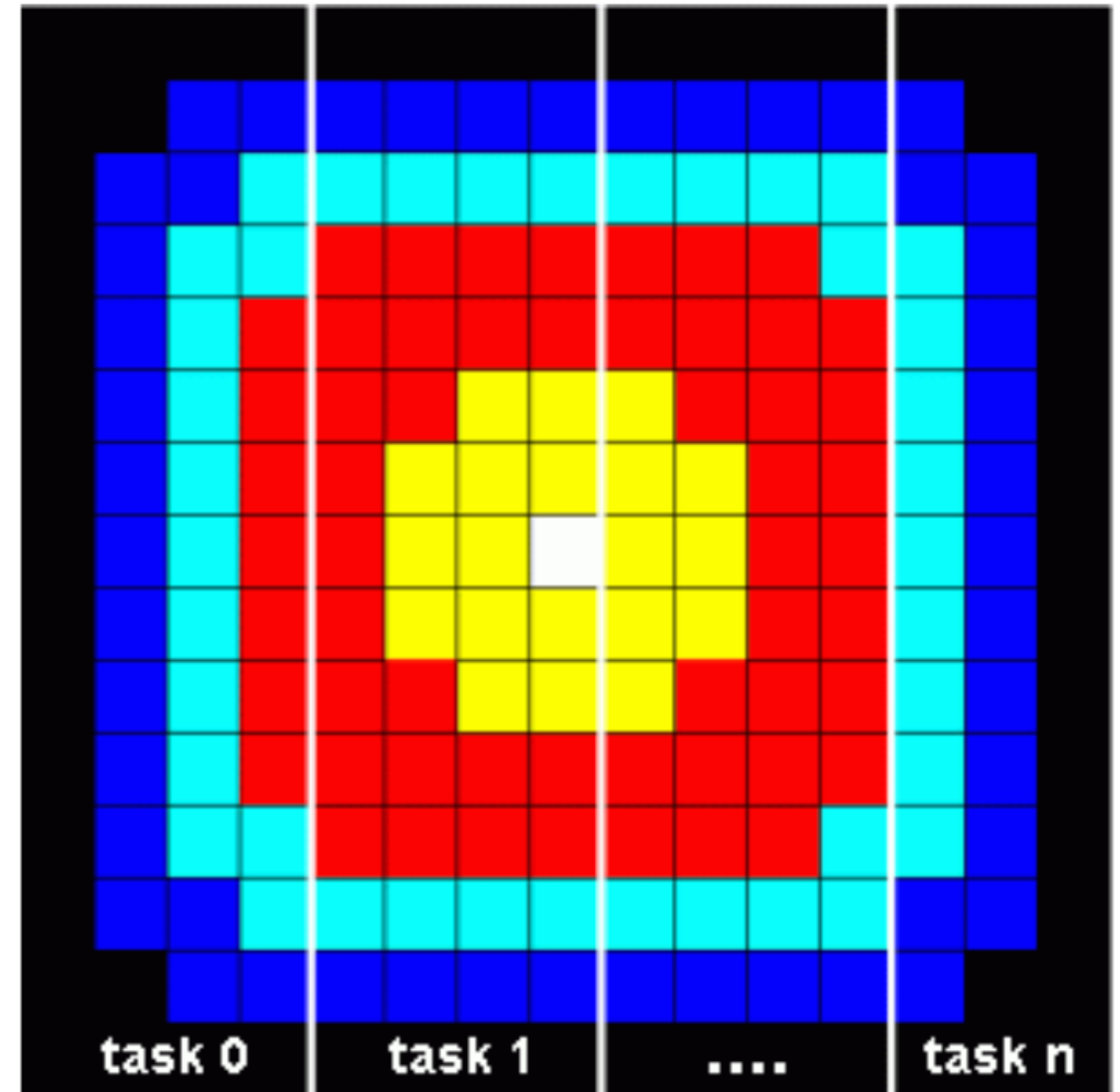


MÉCANISMES DE SYNCHRONISATION

MUTEX (EXCLUSION MUTUELLE)	UN VERROU (MUTEX) QUI PERMET À UN SEUL THREAD OU PROCESSUS D'ACCÉDER À UNE RESSOURCE PARTAGÉE À LA FOIS.
SÉMAPHORE	UNE GÉNÉRALISATION DU MUTEX PERMETTANT À UN NOMBRE FIXE DE THREADS OU PROCESSUS D'ACCÉDER SIMULTANÉMENT À UNE RESSOURCE PARTAGÉE.
VARIABLES CONDITIONNELLES	PERMETTENT À UN THREAD D'ATTENDRE QU'UNE CONDITION SPÉCIFIQUE DEVIENNE VRAIE AVANT DE POURSUIVRE SON EXÉCUTION.
BARRIÈRE	UN POINT DE SYNCHRONISATION QUI OBLIGE LES THREADS OU PROCESSUS À ATTENDRE QUE TOUS LES PARTICIPANTS AIENT ATTEINT LA BARRIÈRE AVANT DE POUVOIR CONTINUER.
VERROU LECTEUR-ÉCRIVAIN	PERMET À PLUSIEURS LECTEURS D'ACCÉDER SIMULTANÉMENT À UNE RESSOURCE PARTAGÉE, MAIS GARANTIT UN ACCÈS EXCLUSIF POUR LES ÉCRIVAINS.
OPÉRATIONS ATOMIQUES	DES OPÉRATIONS DE BAS NIVEAU GARANTISSANT L'EXÉCUTION SANS INTERRUPTION PAR D'AUTRES THREADS OU PROCESSUS.

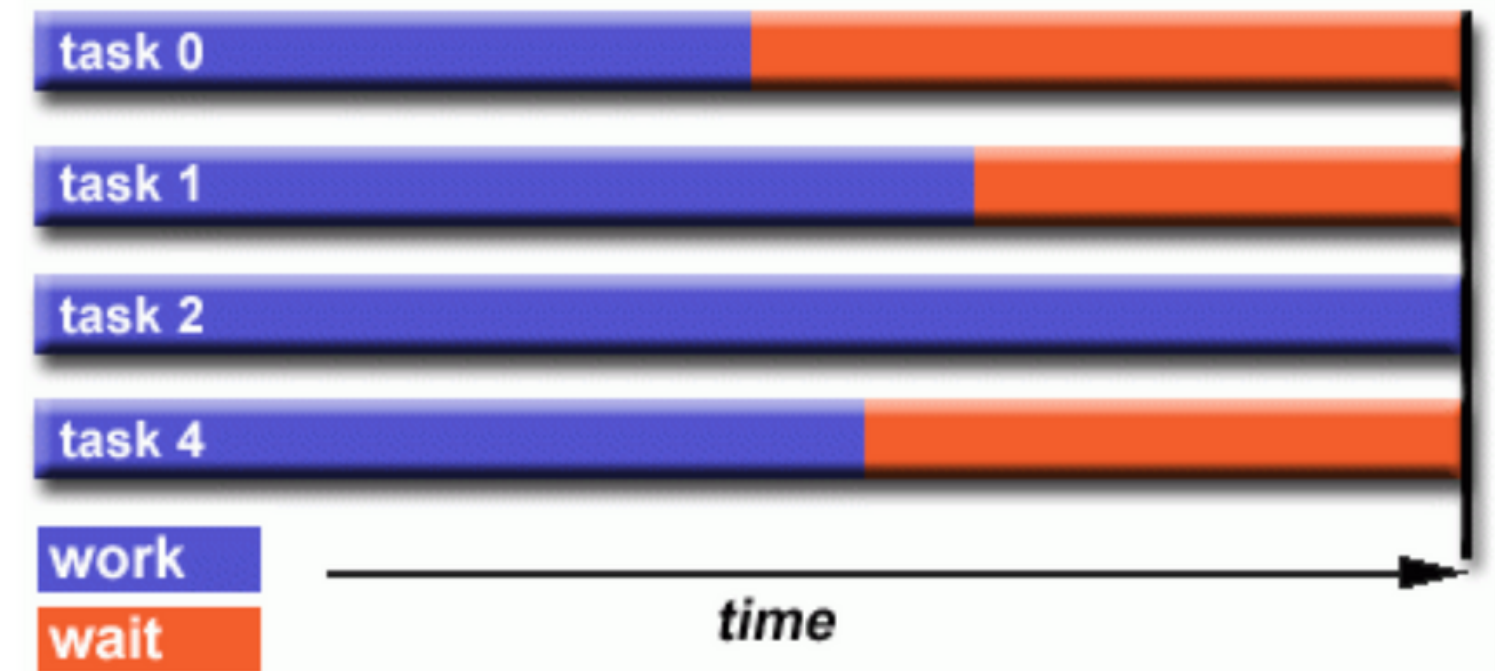
DÉPENDANCE DE DONNÉES

- Dépendance entre les instructions d'un programme lorsque l'ordre d'exécution des instructions affecte les résultats du programme.
- Une dépendance de données résulte de l'utilisation multiple du ou des mêmes emplacements de stockage par différentes tâches.
- Les dépendances sont importantes pour la programmation parallèle car elles constituent l'un des principaux obstacles au parallélisme.



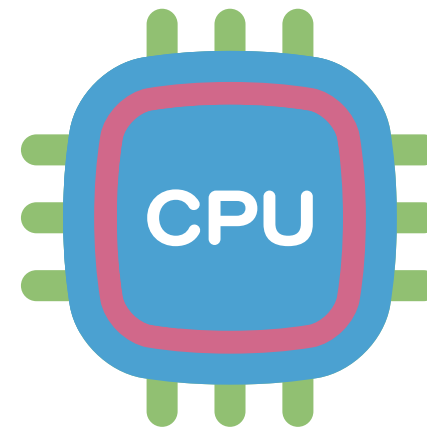
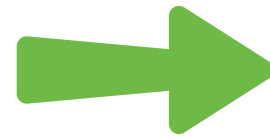
ÉQUILIBRAGE DE LA CHARGE

- Consiste à répartir des quantités de travail à peu près égales entre les tâches de manière à ce que toutes les tâches soient occupées en permanence.
- On peut considérer qu'il s'agit d'une minimisation du temps d'inactivité des tâches.
- Important pour les programmes parallèles pour des raisons de performance. Par exemple, si toutes les tâches sont soumises à un point de synchronisation de barrière, la tâche la plus lente déterminera la performance globale.

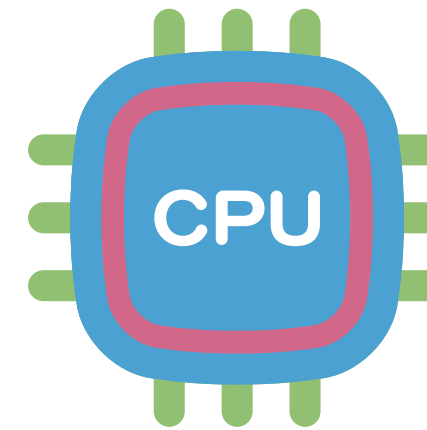


ÉQUILIBRAGE DE LA CHARGE

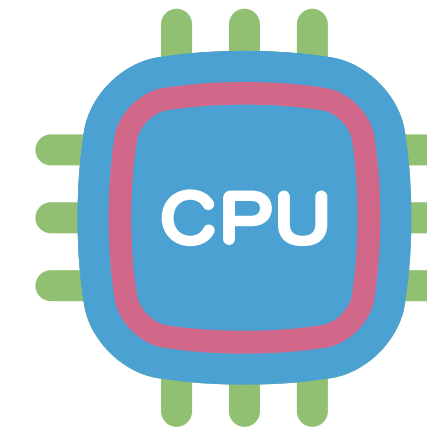
Exemple
simple



CAPACITÉ
1 INS/SEC

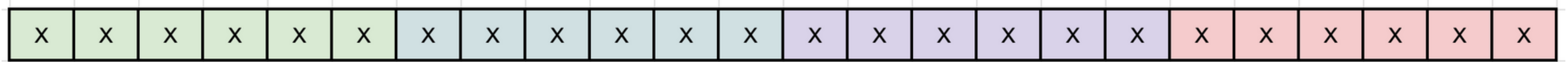


2 INS/SEC

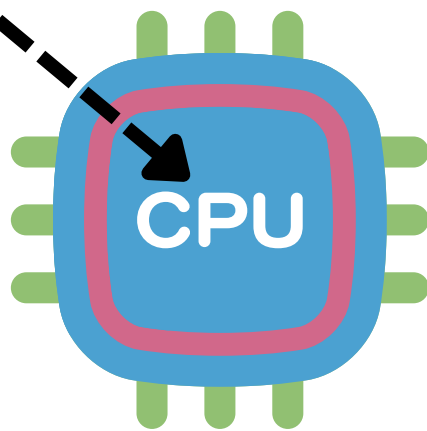


1 INS/SEC

TABLEAU DE 24 ÉLÉMENTS

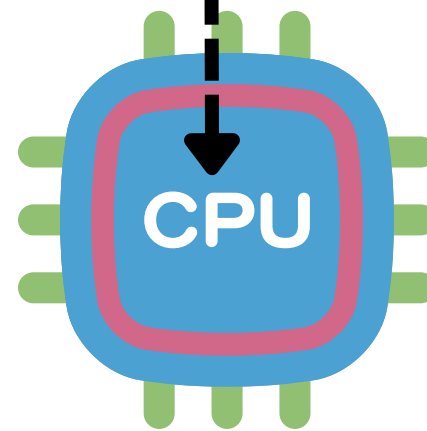


Répartition de tableau
entre différents
processeurs de capacité
différentes

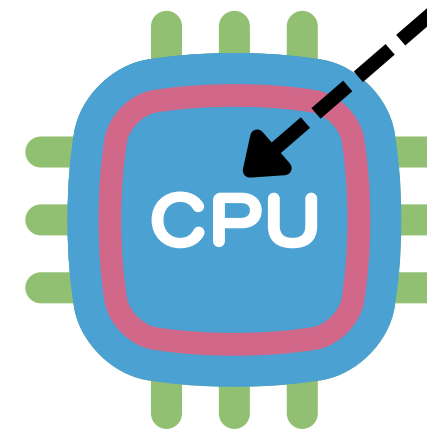


CAPACITÉ

1 INS/SEC



2 INS/SEC

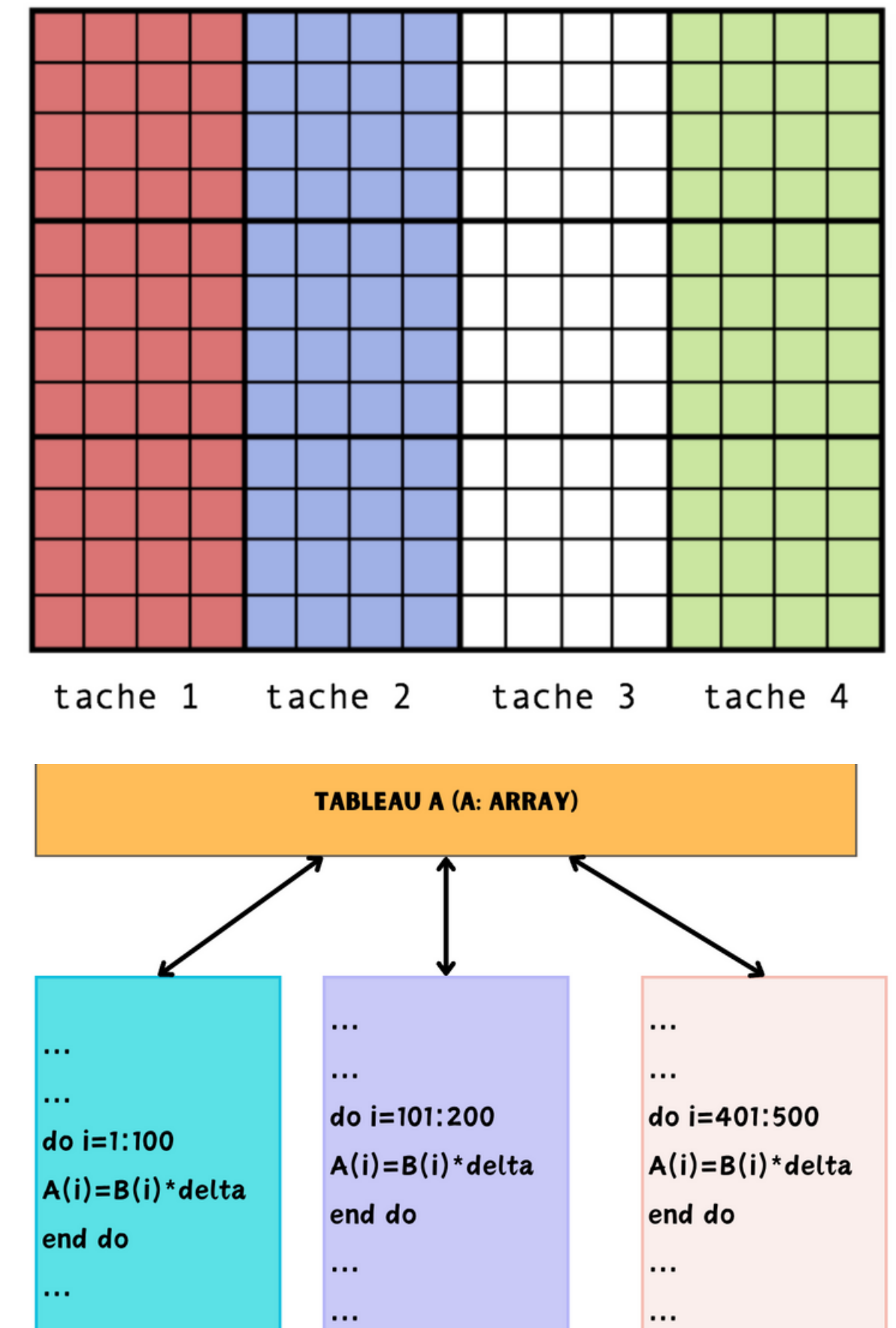


1 INS/SEC

ÉQUILIBRAGE DE LA CHARGE

Comment équilibrer la charge

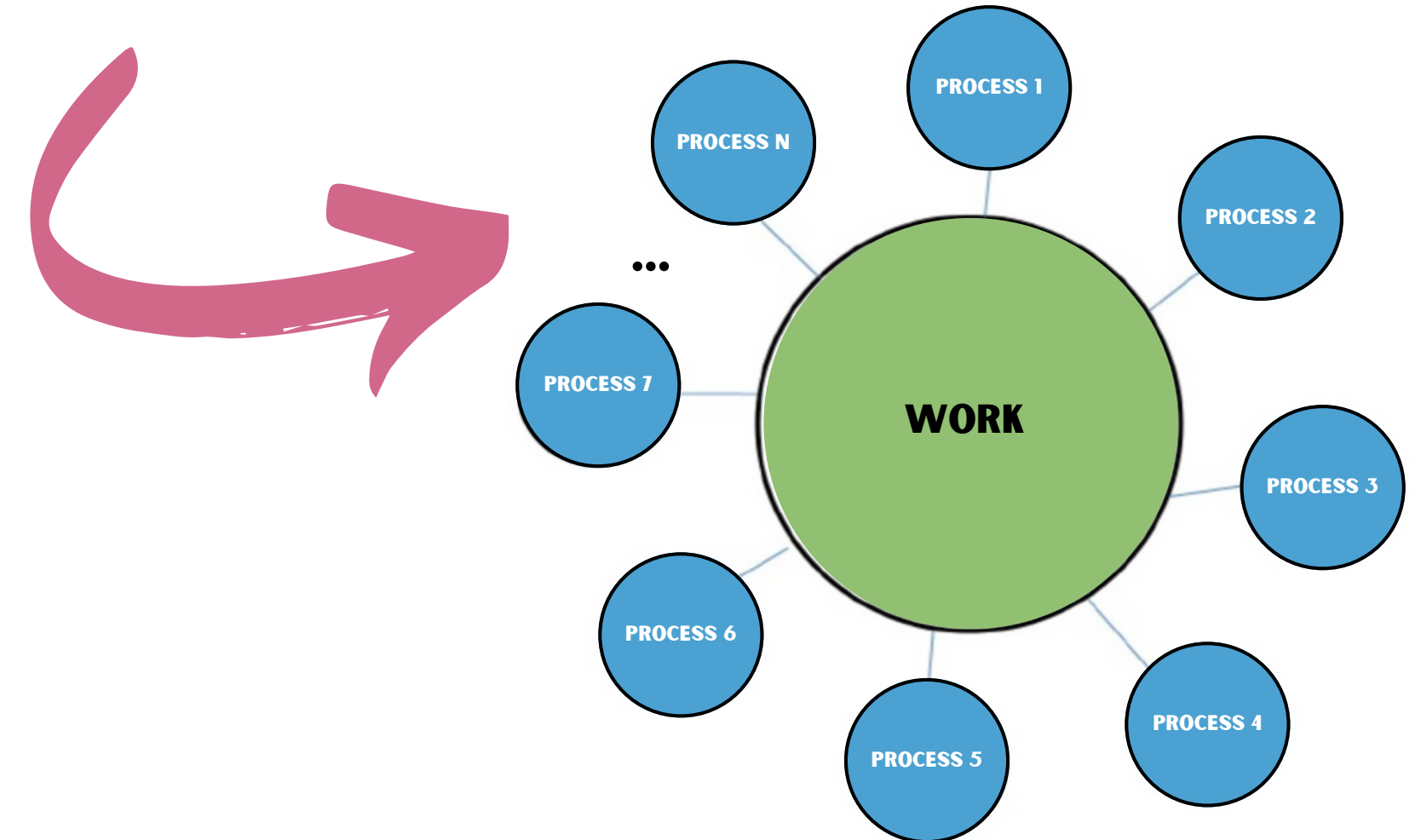
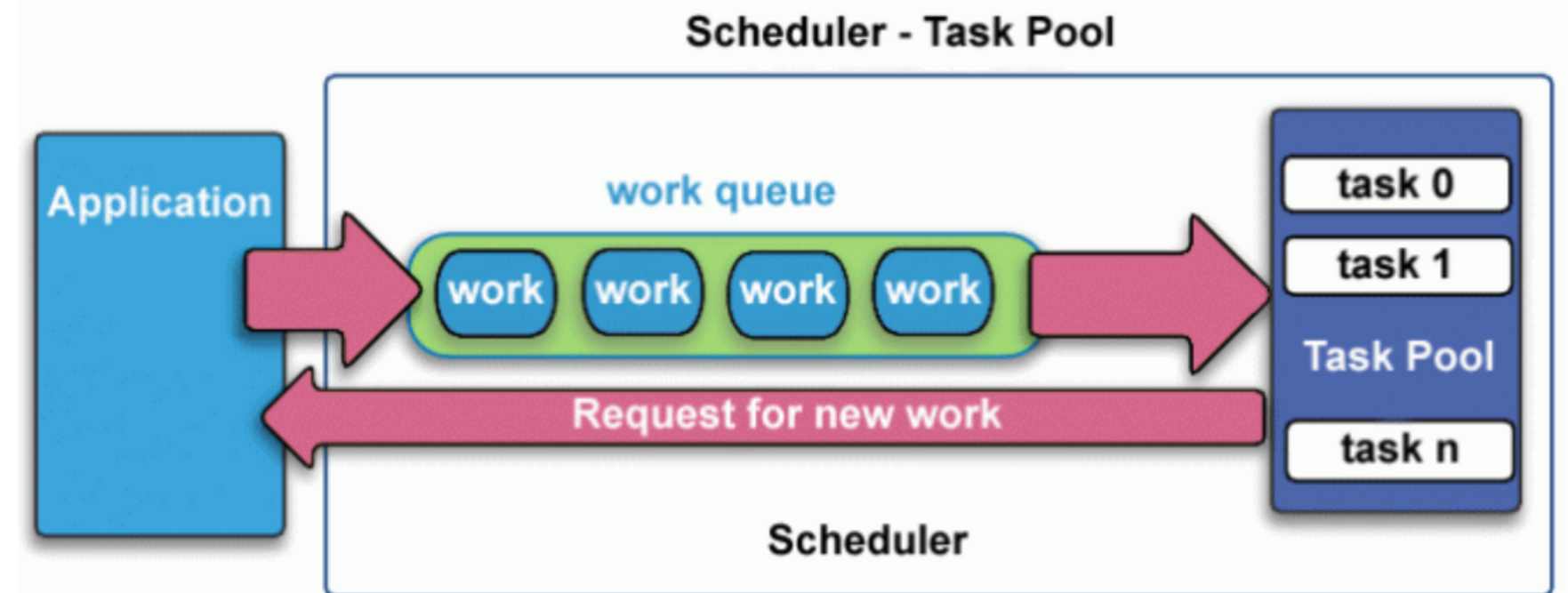
- Pour les tableaux et les matrices où chaque tâche effectue un travail similaire, répartissez uniformément l'ensemble des données entre les tâches.
- Pour les itérations de boucle où le travail effectué à chaque itération est similaire, répartir également les itérations entre les tâches.



ÉQUILIBRAGE DE LA CHARGE

Affectation dynamique du travail

- Certaines catégories de problèmes entraînent des déséquilibres de charge même si les données sont réparties uniformément entre les tâches
- Lorsque la quantité de travail que chaque tâche doit effectuer est intentionnellement variable ou ne peut être prédite, il peut être utile d'utiliser une approche de type "scheduler-task pool".
- Lorsque chaque tâche termine son travail, elle reçoit une nouvelle pièce de la file d'attente.

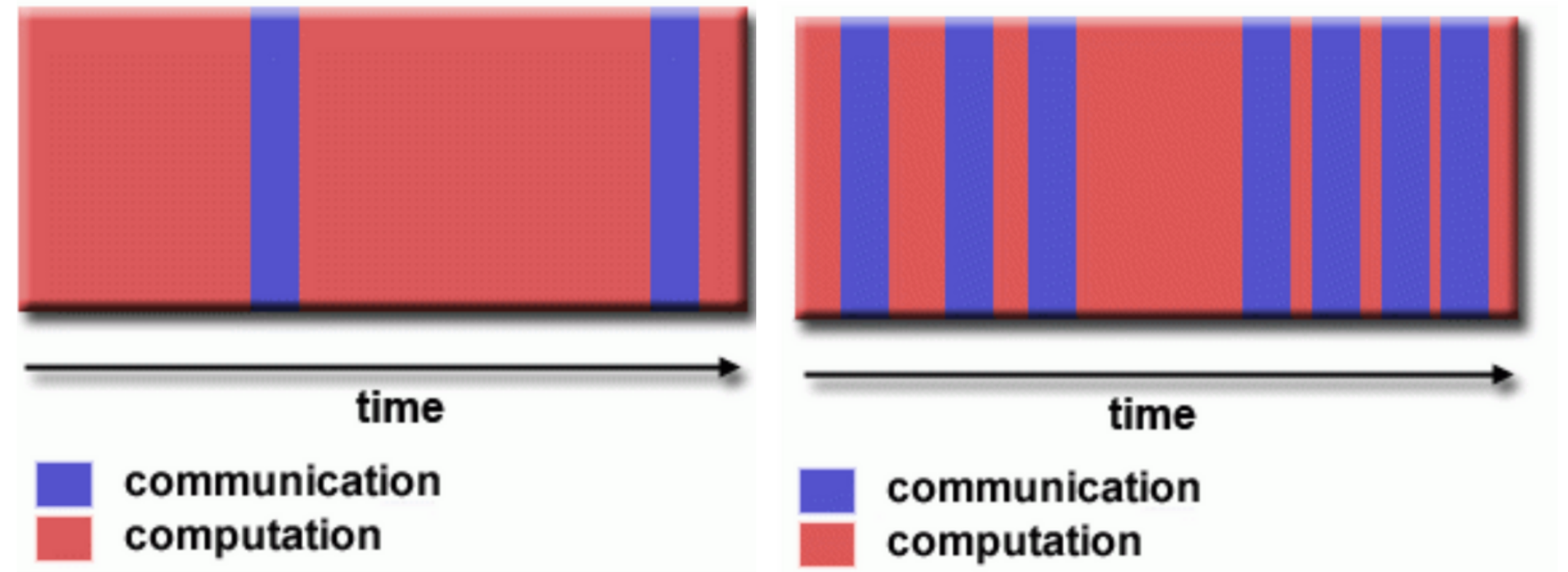


Granularité

RAPPORT ENTRE CALCUL - COMMUNICATION

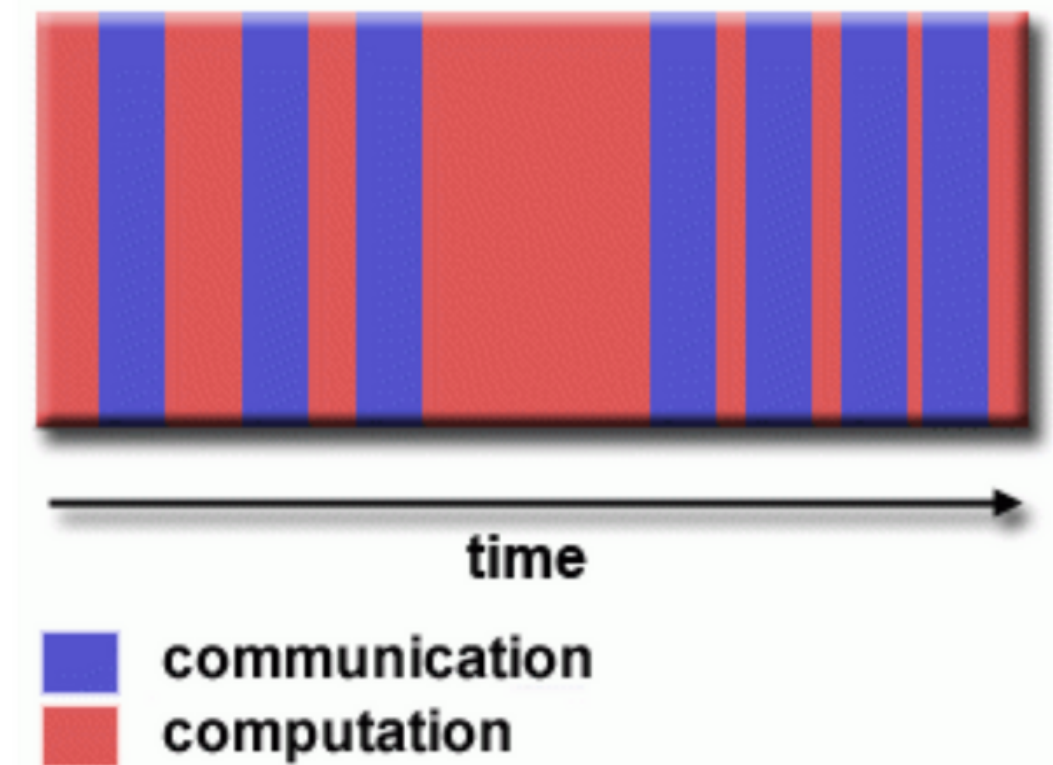
En informatique parallèle, la **granularité** est une mesure qualitative du rapport entre le calcul et la communication.

Les périodes de calcul sont généralement séparées des périodes de communication par des événements de synchronisation.



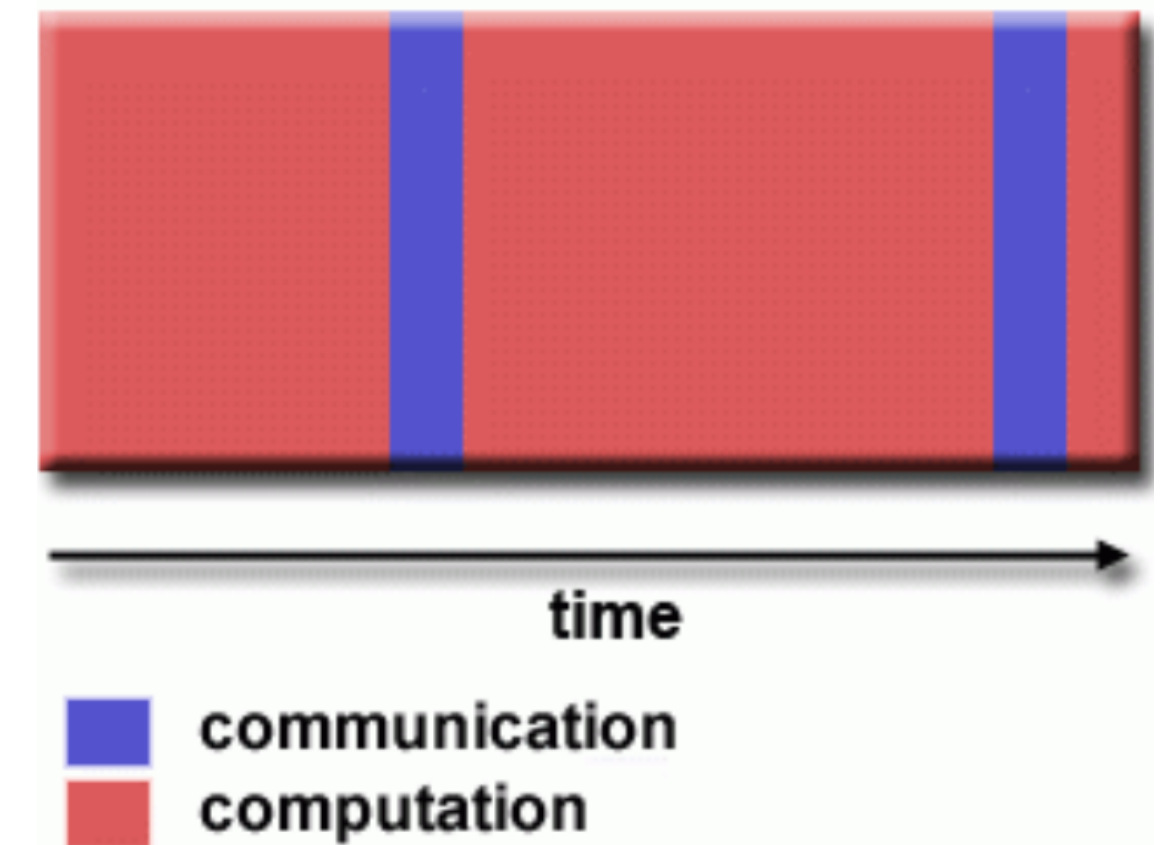
PARALLÉLISME À GRANULARITÉ FINE

- Des quantités relativement faibles de travail de calcul sont effectuées entre les événements de communication.
- Faible ratio calcul/communication.
- Facilite l'équilibrage des charges.
- Implique une surcharge de communication élevée et moins de possibilités d'amélioration des performances.
- Si la granularité est trop fine, il est possible que les coûts généraux requis pour les communications et la synchronisation entre les tâches prennent plus de temps que le calcul.



PARALLÉLISME À GRANULARITÉ ÉLEVÉE

- Des quantités relativement importantes de travail de calcul sont effectuées entre les événements de communication/synchronisation.
- Rapport calcul/communication élevé
- Implique plus d'opportunités d'augmentation des performances
- Il est plus difficile d'équilibrer efficacement la charge



QUELLE EST LA MEILLEURE SOLUTION ?

- **La granularité la plus efficace dépend de l'algorithme et de l'environnement matériel dans lequel il s'exécute.**
- **Dans la plupart des cas, les coûts généraux associés aux communications et à la synchronisation sont élevés par rapport à la vitesse d'exécution, de sorte qu'il est avantageux d'avoir une granularité élevée.**
- **Le parallélisme à graines fines peut contribuer à réduire les frais généraux dus au déséquilibre de la charge.**

Entrées/Sorties

ENTRÉES/SORTIES VS PARALLÉLISME

- **Les opérations d'E/S sont généralement considérées comme des obstacles au parallélisme.**
- **Les opérations d'E/S prennent beaucoup plus de temps que les opérations de mémoire.**
- **Les systèmes d'E/S parallèles peuvent être immatures ou ne pas être disponibles pour toutes les plates-formes.**
- **Dans un environnement où toutes les tâches voient le même espace de fichier, les opérations d'écriture peuvent entraîner l'écrasement du fichier.**
- **Les opérations de lecture peuvent être affectées par la capacité du serveur de fichiers à traiter simultanément plusieurs demandes de lecture.**
- **Les E/S qui doivent être effectuées sur le réseau (NFS, non local) peuvent provoquer d'importants goulets d'étranglement et même faire planter les serveurs de fichiers.**

SYSTÈMES DE FICHIERS PARALLÈLES

- **GPFS : General Parallel File System (IBM) (IBM Spectrum Scale)**
- **Lustre : pour les clusters Linux (Intel)**
- **HDFS : système de fichiers distribués Hadoop (Apache)**
- **PanFS : Panasas ActiveScale File System pour les grappes Linux (Panasas, Inc.)**
- **La spécification de l'interface de programmation d'E/S parallèles pour MPI est disponible depuis 1996 dans le cadre de MPI-2. Des implémentations de fournisseurs et des implémentations "libres" sont maintenant couramment disponibles.**

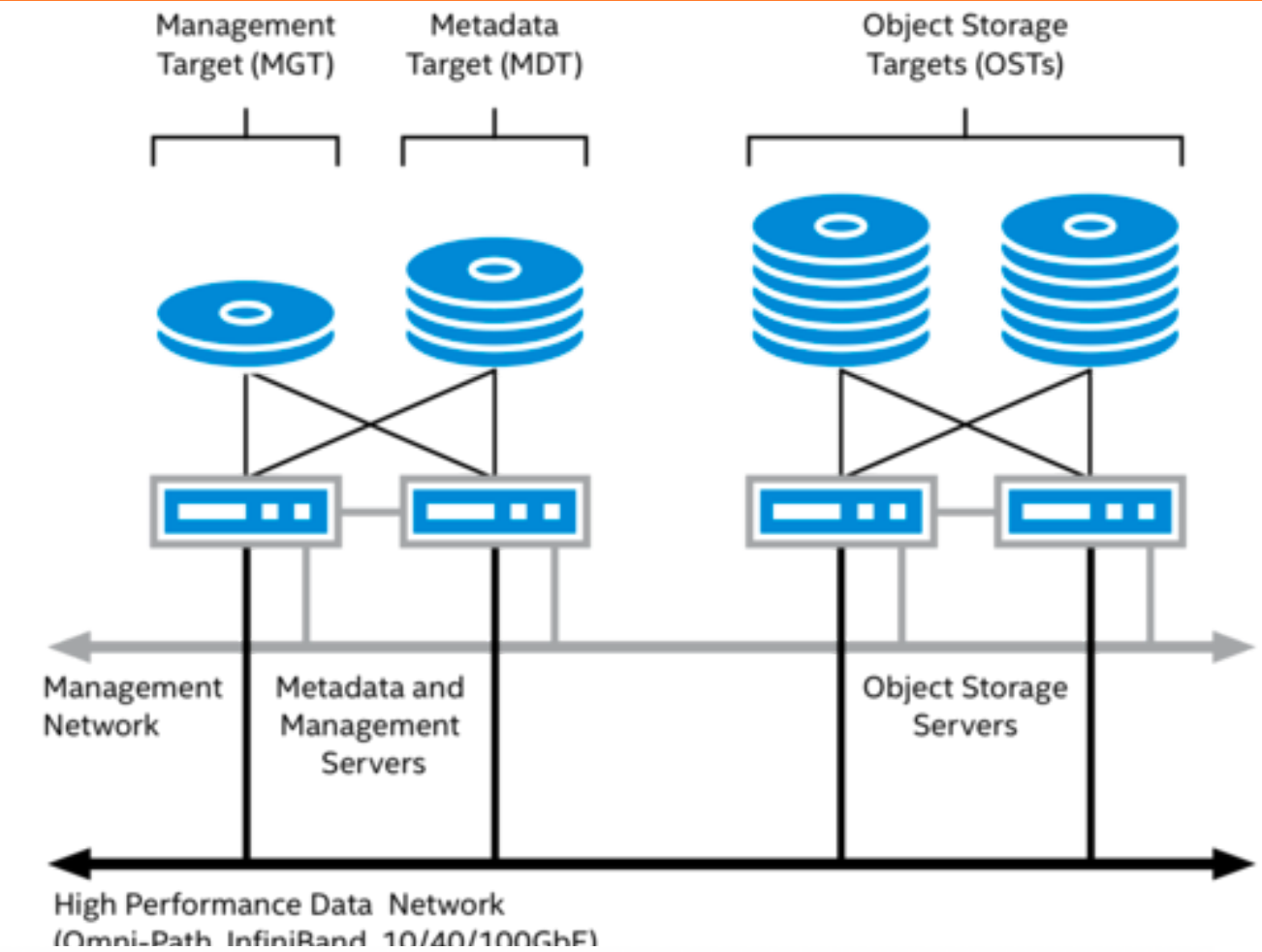
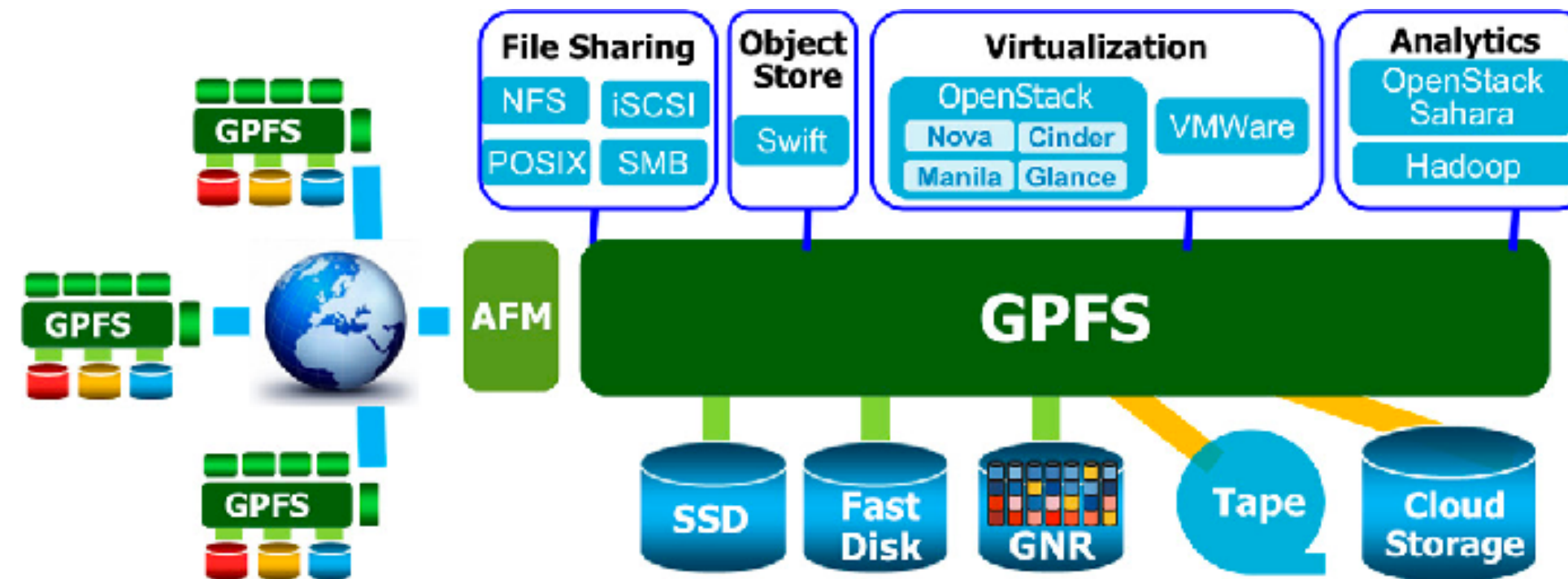


List of file systems

The following lists identify, characterize, and link to more thorough information on file systems. Many older operating systems support only their one...

W Wikipedia / 12:38 PM

SYSTÈMES DE FICHIERS PARALLÈLES



HDFS Architecture

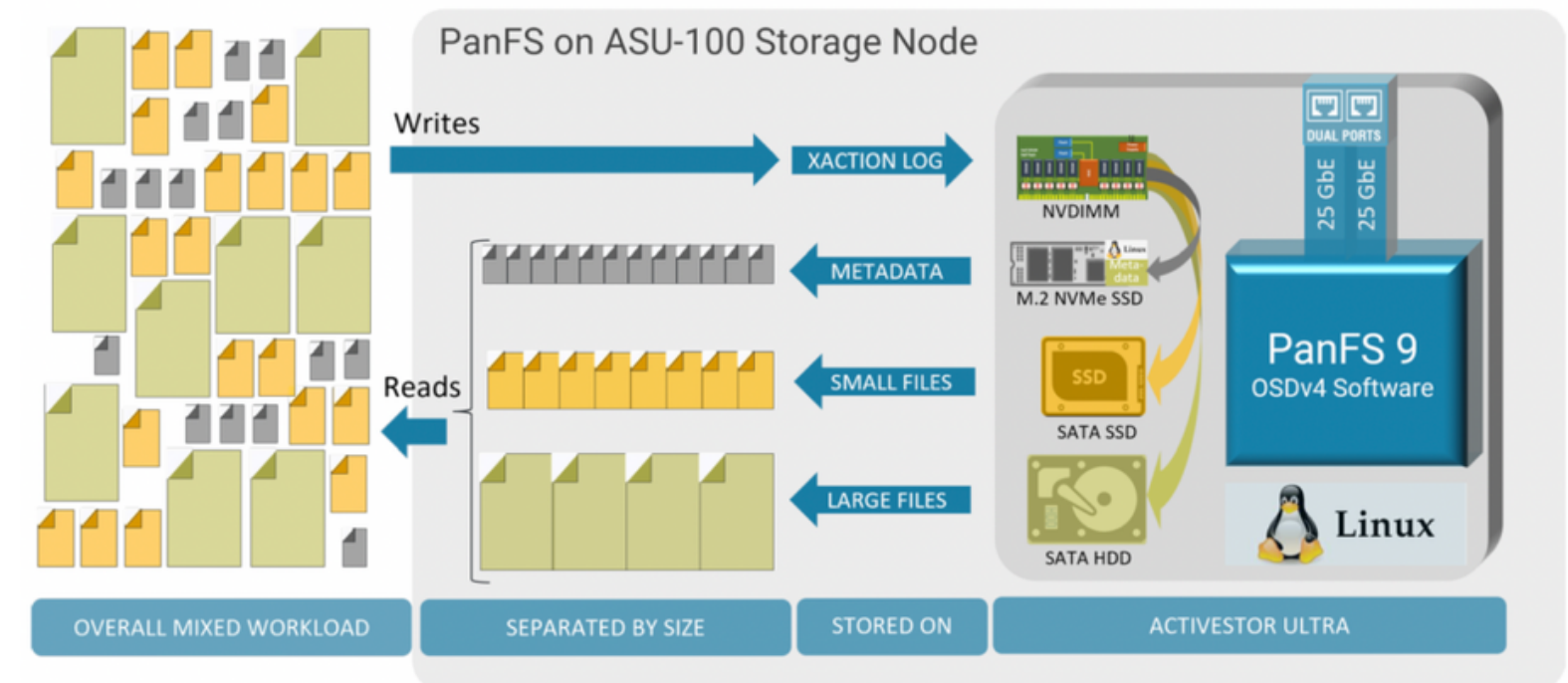
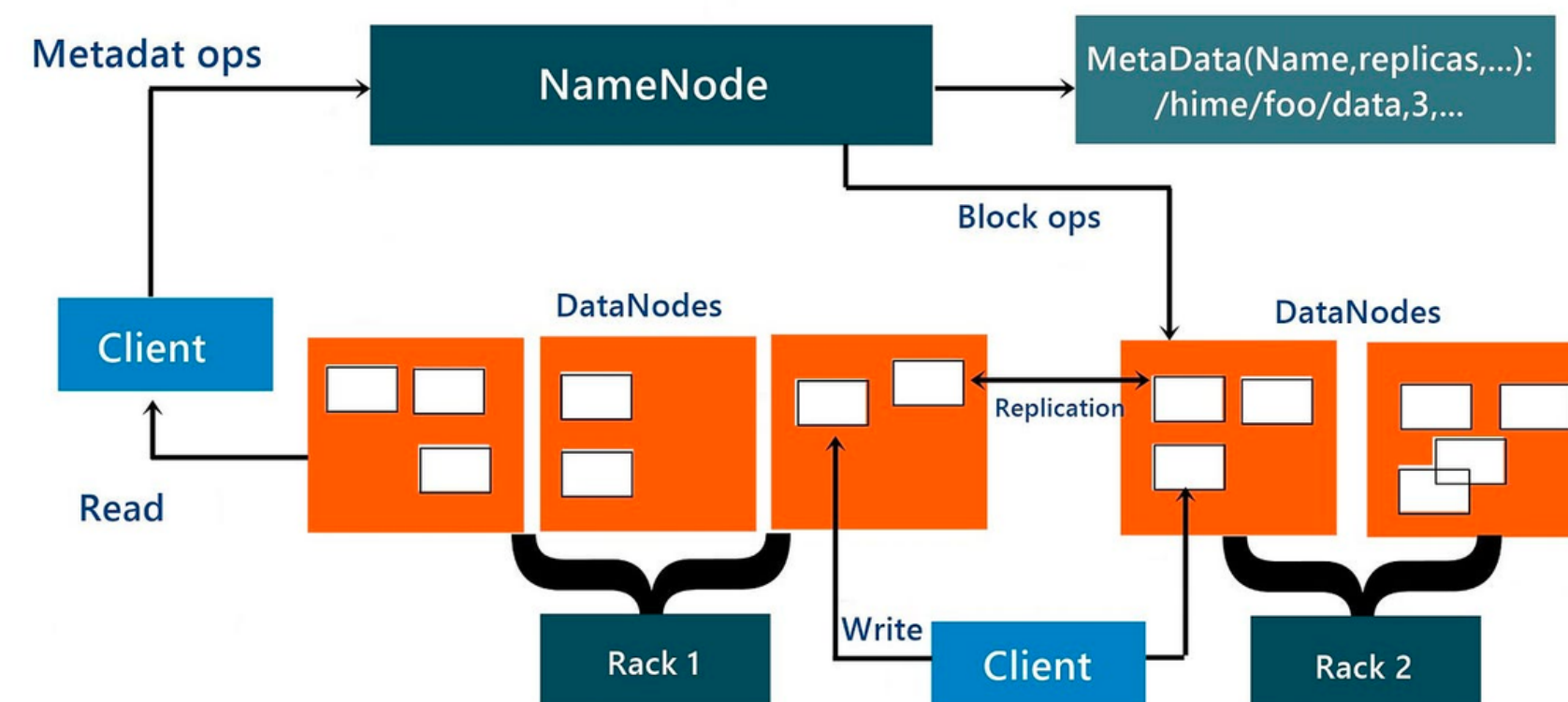


Figure 2. PanFS Dynamic Data Acceleration illustrated on an ASU-100.

BONNES PRATIQUES POUR ÉVITER LES E/S

- **Important! Réduire autant que possible les E/S globales.**
- **Si vous avez accès à un système de fichiers parallèle, utilisez-le.**
- **L'écriture de gros morceaux de données plutôt que de petits morceaux est généralement beaucoup plus efficace.**
- **Un nombre réduit de fichiers de grande taille est plus efficace qu'un grand nombre de petits fichiers.**

BONNES PRATIQUES POUR ÉVITER LES E/S

- **Limitez les E/S à des parties sérielles spécifiques de la tâche, puis utilisez les communications parallèles pour distribuer les données aux tâches parallèles. Par exemple, la tâche 1 peut lire un fichier d'entrée, puis communiquer les données requises aux autres tâches. De même, la tâche 1 pourrait effectuer une opération d'écriture après avoir reçu les données requises de toutes les autres tâches.**
- **Agréger les opérations d'E/S entre les tâches - plutôt que de faire exécuter les E/S par de nombreuses tâches, demander à un sous-ensemble de tâches de les exécuter.**