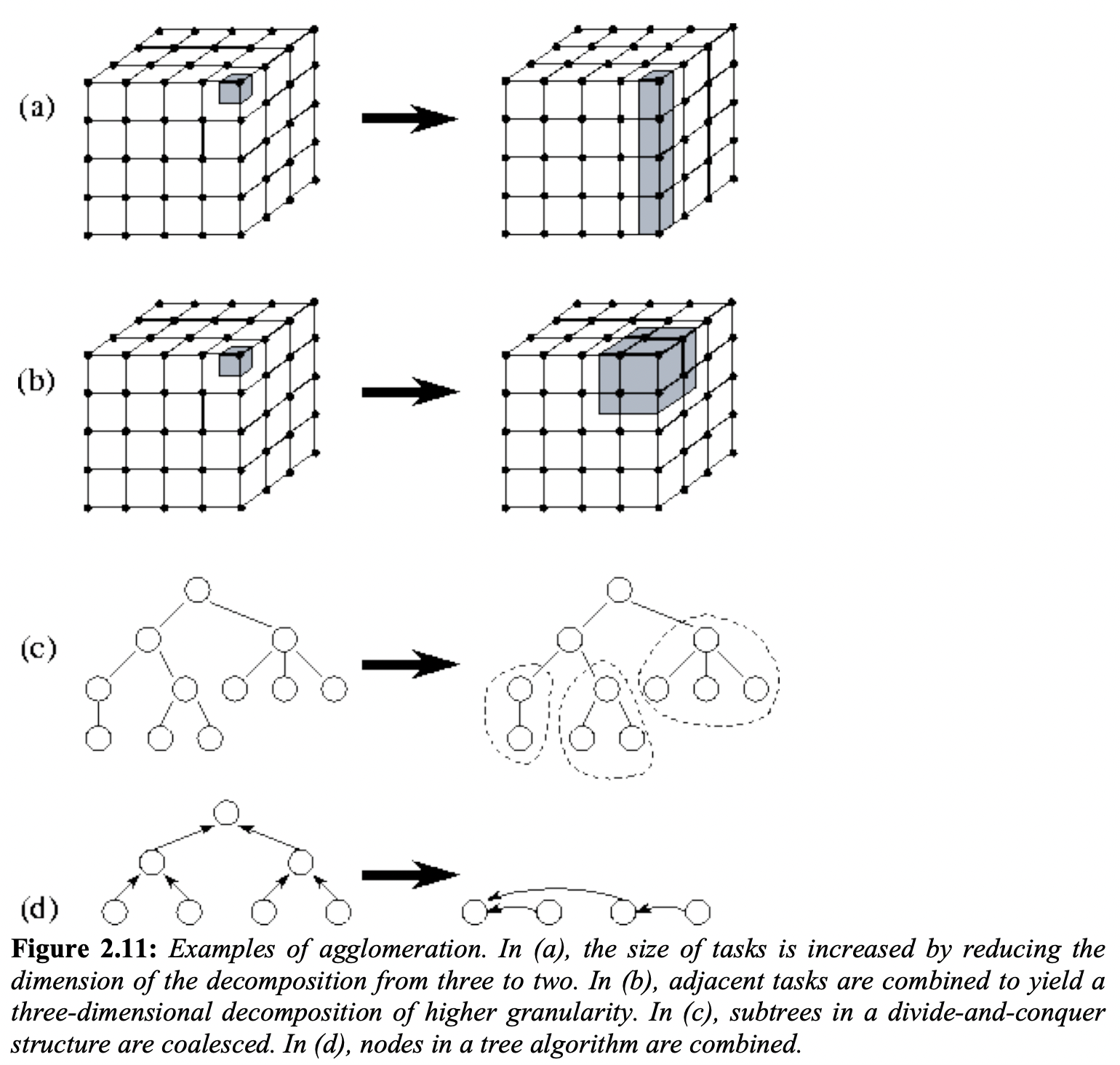
# **Exercice 8. Agglomération et Optimisation de la Concurrence**

#### **Objectif :** Comprendre et appliquer la phase d’agglomération dans la conception d’algorithmes parallèles afin d’améliorer l’efficacité de l’exécution sur une architecture parallèle spécifique.



Vous devez optimiser un algorithme parallèle en appliquant la technique d’**agglomération**. L’algorithme initial suit une approche naïve où chaque unité de calcul est traitée indépendamment, générant ainsi un nombre excessif de tâches.

Votre objectif est de **regrouper** les tâches de manière stratégique afin de :

1. **Réduire la surcharge de communication** en augmentant la granularité des tâches.
2. **Maximiser l’efficacité sur un nombre donné de processeurs**.
3. **Assurer une bonne évolutivité** en permettant une répartition flexible des tâches.

### **Cas d’étude : Calcul de la somme d’un grand tableau en parallèle**

1. **Approche initiale (partitionnement de base)**
   * Le tableau est divisé en *N* parties, chaque élément étant traité séparément.
   * Chaque tâche effectue une somme locale et envoie le résultat à une tâche maître qui effectue la somme finale.
2. **Problème identifié :**
   * Communication excessive entre les tâches et le maître.
   * Séquentialisation de la somme finale.
   * Mauvaise répartition du travail si le nombre de tâches dépasse celui des processeurs disponibles.
3. **Optimisation par agglomération :**Groupez plusieurs éléments du tableau dans des sous-blocs plus grands.
   * Attribuez chaque sous-bloc à un processeur, qui calcule une somme partielle.
   * Effectuez une **réduction en arbre** (tree reduction) pour réduire le nombre d’étapes.
   * Comparez une approche avec et sans réplication des données.

### **Instructions :**

1. Implémentez une version naïve de la somme parallèle.
2. Optimisez cette version en appliquant l’agglomération et en limitant les communications inutiles.
3. Expérimentez avec différents nombres de tâches et de processeurs.
4. Analysez l’impact sur le temps d’exécution et la communication.

### **Questions :**

1. Quelle est la réduction du nombre de communications après agglomération ?
2. Comment choisir la taille optimale d’une tâche après regroupement ?
3. Comment l’agglomération affecte-t-elle l’évolutivité de l’algorithme ?
4. Peut-on appliquer cette technique à d’autres algorithmes de calcul parallèle ?

🚀 **Bonus :** Implémentez l’algorithme en utilisant **MPI** (mpi4py) et comparez les performances avant et après l’agglomération !