

# Stockage de Données Personnelles

January 2020

## 1 Contexte

De nombreuses applications à échelle mondiale manipulent des “données personnelles”, de taille variable. Une donnée est dite personnelle si elle n’est accédée que par un seul utilisateur. Afin de fournir un service efficace, le système de stockage dispose de différent noeud système et doit placer les données sur ces différents noeuds.

Chaque noeud système peut recevoir une ou plusieurs données en fonction de leur taille. La capacité de stockage d’un tel noeud lui est propre et peut différer selon les noeuds. De plus, un noeud système peut communiquer avec tous les autres noeuds système et certains utilisateurs.

Les utilisateurs sont intéressés par une liste de données et accèdent uniquement à ces données. Ils peuvent communiquer directement avec un seul noeud système et ne peuvent pas communiquer avec les autres utilisateurs.

**note:** Dans un tel contexte, le temps de communications entre deux noeuds système ou un noeud système et un utilisateur diffère selon les utilisateurs et noeuds système concernés. Par souci de simplification on considèra que le temps de communication entre deux noeuds  $n_i, n_j$  est constant, mais différent du temps de communication d’un autre couple :  $n_i, n_k$ .

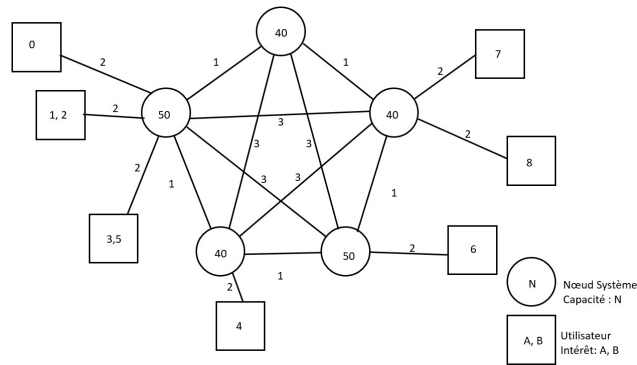


Figure 1: Modélisation d’un système

## 2 Projet

1. Concevoir et implémenter les structures de données pour modéliser les données personnelles, les noeuds système et les utilisateurs. La structure devra, entre autres, implémenter les champs suivants:
  - Données
    - ID : Un entier unique
    - Taille : Un entier
  - Utilisateurs
    - ID : Un entier unique
    - Liste de données d'intérêt : Une liste d'ID (données)
    - Noeud système accessible : Un ID (noeud système)
  - Noeuds système:
    - ID : Un entier unique
    - capacité mémoire : Un entier
    - Liste des données stockées localement : Une liste d'ID (données)
    - Liste des noeuds accessibles : Une liste d'ID (noeud système/Utilisateur)

D'autres caractéristiques pourront être ajoutées pour faciliter le développement du programme.
2. On cherche à trouver un placement efficace pour les données du système: chaque donnée doit être placée au plus près de l'utilisateur intéressé par cette donnée. Implémentez une méthode permettant au système de savoir où placer les données du systèmes. Cette méthode devra prendre en compte la place restante sur les différents noeuds du système. Dans cette méthode, on s'intéressera à placer les données une par une, par ordre d'ID.

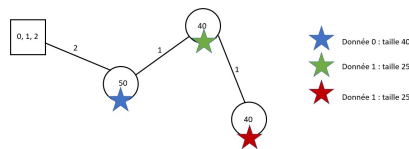


Figure 2: Exemple de placement

- Adaptez votre programme pour que deux utilisateurs puissent être intéressés par une même donnée. On pourra, par exemple, calculer le noeud système a recevoir la donnée pour le premier utilisateur puis calculer le plus chemin entre ce noeud et le second utilisateur. Il faudra alors choisir le noeud de ce chemin qui minimise le temps d'accès pour les deux utilisateurs.

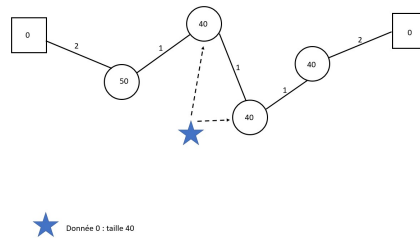


Figure 3: Exemple de placement avec plusieurs intéressés

- Le placement de donnée une par une pouvant mener à des résultats non-optimaux, il convient d'utiliser des algorithmes permettant de repartir efficacement l'espace disponible. Ce problème est connu sous le noms de "Sac à dos Multiple (MKP problem)". Implémentez une solution à ce problème dans le cadre du stockage de données personnelles.

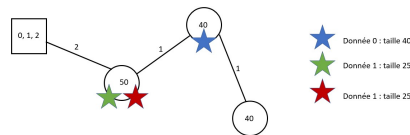


Figure 4: Exemple de placement optimisé