Projet complexité RO :
problème de synchronisation des données

Membres de l'équipe :
Dhouha meliane / Nouha aouachri / Nour douari || **4DS8**

1.1 Description du Problème :

- Il y'a plusieurs systèmes qui contiennent des données
- Ces données doivent être synchronisées entre les systèmes
- Chaque synchronisation prend du temps et des ressources
- Des conflits peuvent survenir lors des synchronisations

1.2 Aspects à minimiser :

- **Le Temps Total de synchronisation (TempsSynchro):**
  Le temps que prend chaque opération de synchronisation

- **Les Coûts des Ressources (CoûtRessources) :**
  Les ressources utilisées (réseau, CPU, mémoire)
  Chaque système a un coût de synchronisation spécifique

- **Les Conflits (ConflitsDonnées):**
  Problèmes qui surviennent quand les données sont différentes entre les systèmes
  Plus il y a de conflits, plus c'est coûteux à résoudre

2. Formulation mathématique du Problème :

➔ **Ensembles:**
- S = {1,...,n} : ensemble des systèmes
- D = {1,...,m} : ensemble des données à synchroniser
- T = {1,...,p} : périodes de temps

➔ **Variables de décision:**

- x[i,j,t] = 1 si la donnée j est synchronisée du système i vers les autres au temps t, 0 sinon
- y[i,j,t] : temps de synchronisation de la donnée j du système i au temps t

➔ **Fonction objectif:**

```
Min Z = Σ(w1*TempsSynchro + w2*CoûtRessources + w3*ConflitsDonnées)

Où:
TempsSynchro = ΣiϵS ΣjϵD ΣtϵT y[i,j,t]
CoûtRessources = ΣiϵS ΣjϵD ΣtϵT (c[i]*x[i,j,t])
ConflitsDonnées = ΣiϵS ΣjϵD ΣtϵT conf[i,j,t]*x[i,j,t]
```

- On cherche à minimiser (Min) la somme pondérée de ces trois composantes on cherche donc à trouver le meilleur compromis entre :

☑ Synchroniser rapidement
☑ Ne pas trop utiliser de ressources
☑ Éviter les conflits de données

- **w1, w2, w3** sont des poids qui permettent de donner plus ou moins d'importance à chaque composante

- **TempsSynchro = Σi∈S Σj∈D Σt∈T y[i,j,t] :**
  Représente le temps total de synchronisation , On somme sur tous les systèmes (S), toutes les données (D) et toutes les périodes (T)
  Exemple : si synchroniser une donnée prend 2 minutes sur le système 1, y[1,1,1] = 2

- **CoûtRessources = Σi∈S Σj∈D Σt∈T (c[i]*x[i,j,t]):**
  Représente le coût total d'utilisation des ressources
  Exemple : si le coût d'utilisation du système 1 est de 10€ et qu'on synchronise (x[1,1,1] = 1), alors le coût est 10€

- **ConflitsDonnées = Σi∈S Σj∈D Σt∈T conf[i,j,t]*x[i,j,t]:**
  Mesure le nombre ou la gravité des conflits de données
  Exemple : si un conflit a une gravité de 5 et qu'on synchronise (x[1,1,1] = 1), alors le coût du conflit est 5

➔ **Contraintes:**

2.1 Contrainte de cohérence :

```
∀i∈S, ∀j∈D, ∀t∈T :
version[i,j,t] = version[k,j,t] ∀k∈S
```

2.2 Contrainte de bande passante:

```
∀t∈T : Σi∈S Σj∈D (taille[j]*x[i,j,t]) ≤ BandePassanteMax
```

2.3 Contrainte de dépendance :

```
∀i∈S, ∀j∈D, ∀t∈T :
x[i,j,t] ≤ x[i,k,t] ∀k∈Dep[j]
```

2.4 Contrainte de fenêtre temporelle :

```
∀i∈S, ∀j∈D, ∀t∈T :
debut[t] ≤ y[i,j,t] ≤ fin[t]
```

3. Les différentes variantes du problémes :

3.1 Synchronisation en temps réel

a. Données mises à jour instantanément.
b. **Contraintes** : Réseau stable, disponibilité continue. Problèmes en cas de panne ou de latence élevée.

3.2 Synchronisation différée

c. Mises à jour périodiques ou déclenchées par des événements.
d. **Contraintes** : Cohérence temporaire affectée, risque de conflits lors des synchronisations.

3.3 Unidirectionnelle

e. Une source unique distribue les mises à jour.
f. **Contraintes** : Moins flexible, mais réduit les conflits.

3.4 Bidirectionnelle

g. Tous les systèmes peuvent se mettre à jour mutuellement.
h. **Contraintes** : Gestion des conflits plus complexe (ex. : CRDT ou règles métiers).

3.5 Avec ou sans gestion des conflits

i. Avec : Résolution automatique ou manuelle des conflits.
j. Sans : Assume qu'il n'y a pas de conflits possibles.
k. **Contraintes** : Nécessite des algorithmes robustes et des règles claires.

2. Chaque variante adapte la synchronisation aux besoins spécifiques, mais impose des compromis selon les contraintes du système.

4. Domaines d'applications :

Systèmes d'information d'entreprise :

- ERP(entreprise resource planning)
- CRM(customer relationship management)
- Gestion des stocks

Cloud Computing :

- Réplication de données
- Sauvegarde distribuée
- Services multicloud

Applications mobiles :

- Mode hors ligne
- Synchronisation multiappareils
- Mise à jour de contenu

Bases de données distribuées :

- Réplication
- Cohérence
- Haute disponibilité

5.Exemple de Résolution - Problème de Synchronisation des Données:

5.1 Données de l'exemple:

Ensembles :

- S = {S1, S2, S3} : 3 systèmes

- D = {D1, D2} : 2 types de données

- T = {T1, T2, T3} : 3 périodes de temps

Paramètres :

- Coûts des ressources par système : c = {S1:10, S2:15, S3:12}

- Taille des données : taille = {D1:5, D2:3}

- Bande passante maximale : BMax = 10

- Poids : w1 = 0.4, w2 = 0.3, w3 = 0.3

5.2 Modélisation mathématique

Variables de décision :

- x[i,j,t] : binaire, = 1 si synchronisation de la donnée j du système i au temps t

- y[i,j,t] : temps de synchronisation de la donnée j du système i au temps t

5.3 Fonction objectif :

Min Z = 0.4*TempsSynchro + 0.3*CoûtRessources + 0.3*ConflitsDonnées

Où :

TempsSynchro = $\Sigma i \in S\ \Sigma j \in D\ \Sigma t \in T$ y[i,j,t]

CoûtRessources = $\Sigma i \in S\ \Sigma j \in D\ \Sigma t \in T$ (c[i]*x[i,j,t])

ConflitsDonnées = $\Sigma i \in S\ \Sigma j \in D\ \Sigma t \in T$ conf[i,j,t]*x[i,j,t]

5.4 Contraintes principales :

- ☑ Contrainte de bande passante :

  $\forall t \in T : \Sigma i \in S\ \Sigma j \in D$ (taille[j]*x[i,j,t]) ≤ 10

- ☑ Contrainte de synchronisation unique :

  $\forall j \in D,\ \forall t \in T : \Sigma i \in S$ x[i,j,t] ≤ 1

5.5 Résolution :

Solution optimale trouvée :

a) Planning des synchronisations (x[i,j,t] = 1) :

- T1 : S1 synchronise D1

- T2 : S2 synchronise D2

- T3 : S3 synchronise D1

b) Valeurs des objectifs :

- TempsSynchro = 15 unités

- CoûtRessources = 37 unités

- ConflitsDonnées = 8 unités

- Z = 0.4*15 + 0.3*37 + 0.3*8 = 6 + 11.1 + 2.4 = 19.5

Vérification des contraintes :

1. Bande passante :

- T1 : 5 ≤ 10 ✓

- T2 : 3 ≤ 10 ✓

- T3 : 5 ≤ 10 ✓

2. Synchronisation unique par période :

- T1 : 1 synchronisation ✓

- T2 : 1 synchronisation ✓

- T3 : 1 synchronisation ✓

5.6 Analyse de la solution

Cette solution :

- Répartit les synchronisations sur les trois périodes

- Respecte les contraintes de bande passante

- Minimise le compromis entre temps, coûts et conflits

- Utilise les systèmes de manière équilibrée

6.Algorithmes proposés pour résoudre le problème étudié:

6.1 Algorithmes Exacts:

1. **Branch and Bound (B&B)**

- **Utilisation** : Pour résoudre le problème de manière optimale
- **Principe** : Diviser l'espace de solutions et éliminer les branches non prometteuses
- **Avantages** : Solution optimale garantie
- **Inconvénients** : Temps de calcul important pour les grandes instances

2.Programmation Dynamique:

- **Utilisation** : Pour les cas où le problème peut être décomposé en sous-problèmes
- **Principe** : Résolution récursive et mémorisation des solutions partielles
- **Avantages** : Efficace pour certaines structures de problèmes
- **Limitations** : Demande beaucoup de mémoire

3. **Décomposition de Benders**

- **Utilisation** : Pour décomposer le problème en sous-problèmes plus simples
- **Principe** : Séparation entre variables primales et duales
- **Avantages** : Peut traiter de grandes instances
- **Application** : Particulièrement utile quand il y a une structure naturelle de décomposition

4. **Algorithme du Simplexe**

- **Utilisation** : Pour la partie linéaire du problème
- **Principe** : Exploration des sommets du polyèdre des solutions
- **Avantages :** Bien étudié et implémenté dans les solveurs

6.2 Algorithmes Approchés :

1. Recherche Tabou:

- **Utilisation** : Pour éviter les optima locaux
- **Composants clés** :
    - Liste taboue
    - Critères d'aspiration
    - Diversification/Intensification

2. Heuristiques Constructives:

- **Principe** : Construction progressive de la solution
- **Types** :
    - Gloutons basés sur le temps
    - Gloutons basés sur les ressources
    - Gloutons basés sur les conflits

3. Méthodes Hybrides:

Combinaison d'algorithmes exacts et approchés

Exemples :

- B&B + Heuristique pour les sous-problèmes
- Recuit simulé + Programmation dynamique
- Algorithme génétique + Recherche locale

6.3 Critères de choix de l'algorithme :

1. Taille du problème
2. Temps de calcul disponible
3. Qualité de solution requise
4. Structure du problème
5. Ressources disponibles

Petites instances → Algorithmes exacts (B&B, Programmation dynamique)

Grandes instances → Métaheuristiques (Algorithmes génétiques, Recuit simulé)

Temps réel → Heuristiques rapides

Calcul offline → Méthodes hybrides