

Rapport de Projet : C-WildWater

Équipe : FREJ Rayene, MONTEIL Maxime, KNOPE Mattéo

Filière : préING2 MI-2

2025-2026

1. Introduction

Le projet **C-WildWater** consiste à développer un système de synthèse de données pour la distribution d'eau en France. L'objectif est de traiter un volume massif de données (plus de 8 millions de lignes) pour générer des histogrammes de performance des usines et calculer les rendements (pertes) du réseau. Pour garantir l'efficacité, nous utilisons une architecture hybride : un script **Shell** pour l'interface et le séquençage, et un programme en **langage C** utilisant des structures **AVL** pour le traitement algorithmique performant.

2. Architecture Technique

Conformément aux exigences, notre application se décompose en trois parties:

- **Script Shell** : Point d'entrée qui valide les arguments, gère la compilation via make et génère les graphiques via **GnuPlot**.
- **Programme C** : Cœur du traitement utilisant des arbres **AVL** pour stocker les usines avec une complexité de recherche logarithmique ($O(\log n)$).
- **Gestion des données** : Lecture d'un fichier CSV de plus de 500 Mo, avec une attention particulière à l'empreinte mémoire et à la libération systématique des ressources (free).

3. Répartition des Tâches (Groupe de 3)

Nous avons divisé le travail pour assurer une progression parallèle:

- **FREJ Rayene ([crayfr](#)) (Responsable Shell & Intégration)** : Développement du script script.sh, gestion des erreurs d'arguments, calcul du temps d'exécution en millisecondes et intégration de GnuPlot pour les histogrammes.
- **MONTEIL Maxime ([MKmatt24](#)) (Responsable Algorithmique AVL)** : Implémentation de la structure AVL générique en C, fonctions d'insertion, de rotation et de recherche pour le traitement histo (volumes max, captés et réels).
- **Knopes Mattéo ([M4x0u77](#)) (Responsable Réseau & Fuites)** : Implémentation du calcul leaks via un arbre de distribution (Stockage -> Jonction -> Raccordement) et gestion de la robustesse de la lecture du fichier CSV.

4. Planning de Réalisation

- **Semaine 1** : Analyse de la structure du fichier CSV et mise en place du Makefile et du squelette du programme C.
- **Semaine 2** : Finalisation de l'AVL et du traitement histo. Premier script Shell fonctionnel.
- **Semaine 3** : Implémentation du parcours d'arbre pour le calcul des pertes (leaks) et tests de performance
- **Semaine 4** : Optimisation mémoire et rédaction du ReadMe et génération des tests finaux.

5. Limitations et Difficultés

- **Gestion des tirets (-)** : La manipulation des valeurs manquantes dans le CSV a nécessité une machine à états pour éviter les erreurs de lecture.
- **Complexité de leaks** : La construction de l'arbre de distribution a été le point le plus complexe, résolu par l'utilisation d'un AVL secondaire pour retrouver rapidement les nœuds parents²⁰.
- **Pas de fonctions bonus** : les fonctions bonus n'ont malheureusement pas pu être implémentées à temps.
- **Gnuplot – Commentaire** : Gnuplot n'arrêtait pas d'interpréter le caractère « # » comme un commentaire, ce qui nous a pris du temps à détecter et à corriger.

6. Conclusion

Ce projet nous a permis de maîtriser l'interaction entre les outils système (Shell/Make) et l'algorithme avancée en C. L'utilisation des structures AVL a été cruciale pour traiter les 500 Mo de données dans un temps raisonnable.