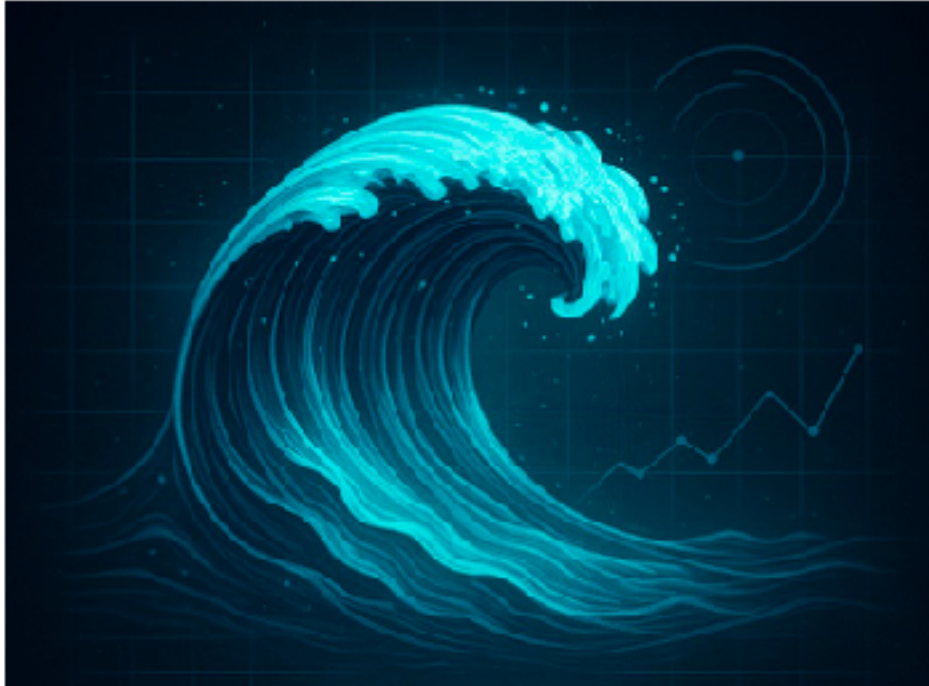


Projet C-Wildwater



AZZI Ikram-Fatima.
DUBLANC Matthieu.
FRIDJINE Ibrahim.

Groupe MI3_I

Rapport de Projet

1. Présentation du projet

Notre projet permet de réaliser un programme permettant de faire la synthèse de données d'un système de distribution d'eau potable. Le programme traite un fichier .csv de données volumineux pour extraire deux types d'informations clés :

- Les statistiques des usines (histo) : Analyse des capacités maximales (max) , des volumes captés aux sources (src) et des volumes réels traités (real) pour générer des histogrammes.
- Le calcul des rendements (leaks) : Identification des fuites en suivant le réseau de l'usine vers l'aval pour calculer la différence entre le volume produit et le volume utilisé.

2. Organisation du groupe

Nous avons opté pour une approche de travail collaboratif. L'essentiel du code a été réalisé ensemble durant les séances de TD, ce qui nous a permis de construire la logique des structures de données en commun. Hors TD, nous avons continué à avancer en groupe plutôt que de diviser les tâches par personne. Cette méthode a été la plus utile pour nous et a permis à ce que chaque membre maîtrise l'ensemble du code.

3. Résumé de notre code

- Partie Shell (myScript.sh) : Point d'entrée du programme. Il gère les arguments, lance la compilation via le Makefile, effectue le filtrage des données et pilote Gnuplot.
- Partie C (main.c, avl_usine.c, leak.c): Assure le traitement lourd des données via des AVL.
- Visualisation (plot_histo.plt) : Script Gnuplot configuré pour générer les histogrammes.

4. État du programme

Le programme est entièrement fonctionnel. Toutes les fonctions déclarées dans avl.h sont implémentées. Le projet compile parfaitement sans erreurs. On respecte la robustesse, par exemple, si une usine est absente du réseau, le programme renvoie bien la valeur -1. La gestion de la mémoire est respectée avec des fonctions de libération (liberer_avl_usine, liberer_graphe_complet) pour éviter toute fuite.

5. Difficultés rencontrées

- Manipulation des fichiers en C : Pour l'ouverture des fichiers, nous avons dû étudier les CM et TD de l'année dernière. Cela nous a permis de manipuler les données correctement pour remplir nos arbres AVL.
- Faibles bases en Shell : Le développement de myScript.sh a été particulièrement difficile. N'ayant pas beaucoup étudié ce langage en TD, nous n'avions pas d'aussi bonnes bases qu'en langage C. Nous avons rencontré quelques problèmes pour filtrer les données et obtenir le nombre exact d'usines pour les histogrammes.

- Calcul des fuites : En effet le code a nécessité de nombreux ajustements pour s'assurer que les pertes cumulées étaient exactes.

6. Conclusion

Ce projet nous a permis de mobiliser nos connaissances sur des cas d'usage réels. Malgré les difficultés rencontrées, nous avons réussi à produire un outil robuste, performant et parfaitement fonctionnel en respectant toutes les consignes.