Wael W.
Hatim M.
Yanis A.

MI-1J

Rapport de Projet : C-WildWater

1. Introduction

Le projet C-WildWater consiste à développer un système d'analyse pour un réseau de distribution d'eau en France. L'objectif est de traiter un gros volume de données pour générer les histogrammes des usines et calculer les rendements de distribution. Pour cela nous avons utilisé un script Shell pilotant un programme en langage C utilisant des structures de données de type AVL.

2. Organisation et Répartition du Travail

Notre organisation a évolué au cours du projet pour s'adapter aux défis techniques rencontrés :

- Nous avons débuté le développement du script Shell tous ensemble. Cependant, nous avons rapidement réalisé que travailler à trois sur le même script freinait notre progression globale.

- Pour gagner en efficacité, Yanis a pris seul la suite du Shell, tandis que Hatim et Wael ont lancé le développement de la structure en C.

- Yanis ayant rencontré des blocages sur certaines fonctionnalités du Shell, Wael l'a rejoint pour finaliser la partie script. Pendant ce temps, Hatim a poursuivi seul l'implémentation des algorithmes en C.

- Une fois le Shell opérationnel, toute l'équipe s'est réunie pour terminer les dernières fonctions du programme C et assurer la liaison entre les deux langages.

- Hatim s'est chargé de la rédaction du rapport et du fichier README. En parallèle, les autres membres du groupe ont effectué des tests intensifs pour s'assurer que le code s'exécutait sans erreur avant chaque commit et l'envoi final.

3. Difficultés Rencontrées

- La fonctionnalité leaks (calcul des pertes) a été l'obstacle majeur. Contrairement aux histogrammes, il faut parcourir tout le réseau aval d'une usine. Nous avons dû implémenter un arbre à nombre d'enfants qu'on ne connaît pas, couplé à un AVL pour simplifier la recherche. Cette étape a été la plus difficile selon nous.

- L'adaptation constante entre le Shell et le C a été complexe. Comme les deux parties ont souvent été développées séparément, il a fallu une communication permanente pour s'accorder sur le format des données (filtrage CSV en amont) et sur la gestion des codes d'erreur pour que le script Shell puisse décider de la suite du traitement.

- Nous avons peu utilisé les machines de l'école en dehors des séances officielles. Travailler sur nos propres environnements a causé des "frayeurs" lors des séances de TD, où des bugs liés aux spécificités du système de l'école apparaissaient. Nous avons dû gérer ces imprévus sous forte pression temporelle pour garantir un code robuste.

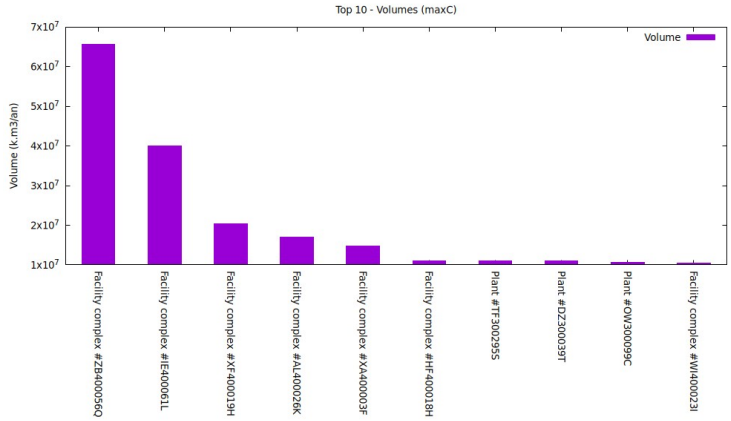
Nous avons peu utilisé les ordinateurs de l'école en dehors des séances de TD. Travailler sur nos propres ordinateurs nous a causé des frayeurs lors des séances de TD car il y a des fois ou le code s'executer parfaitement sur nos machines alors que l'execution ne marchait plus lors des séances de TD (car nos machines sous Windows alors que machines de l'école sous Linux).

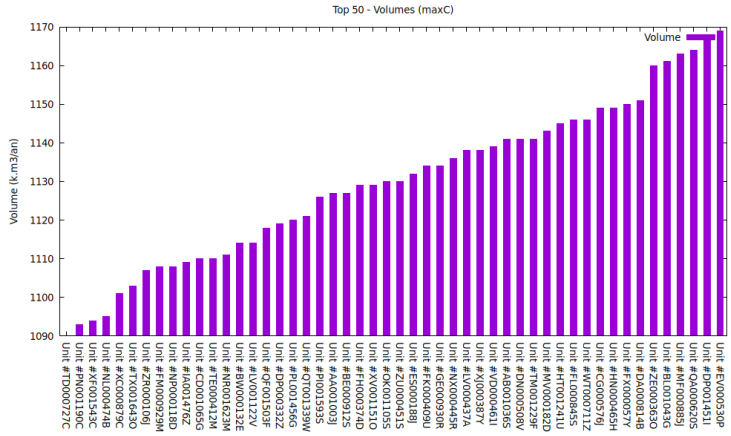
- Nous avons pris la décision de ne pas ajouter les fonctionnalités bonus car le manque de temps et la complexité de la structure de base nous ont poussés à ne pas prendre de risque. Nous avons préféré livrer un code qui ne crash jamais et qui respecte strictement les consignes de base plutôt que de risquer une instabilité du système avec des fonctions supplémentaires.

4. Analyse des Résultats (Histogrammes)

Cette section présente les sorties graphiques générées par notre outil via Gnuplot.

A. Histogramme des Usines (Processus de traitement)



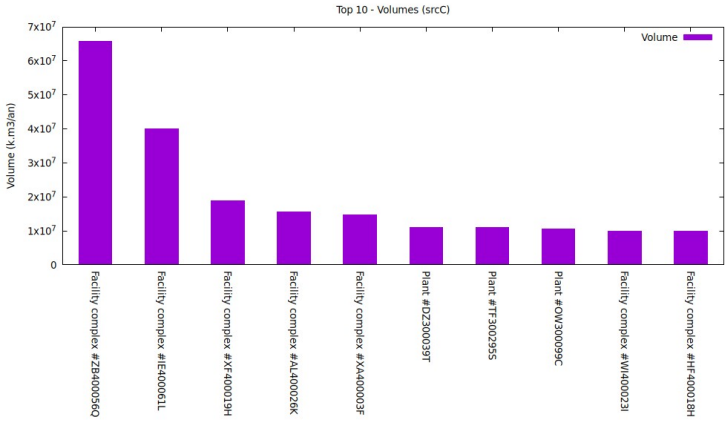


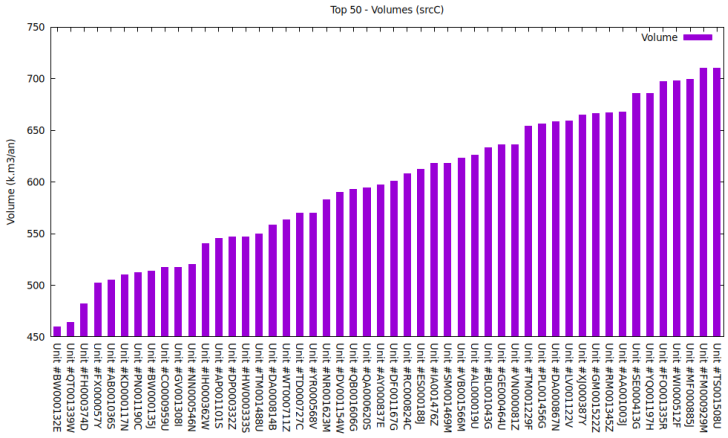
Cet histogramme affiche la capacité totale de traitement de chaque usine de production d'eau par rapport au volume réellement traité.

Axe X : Identifiants des usines.

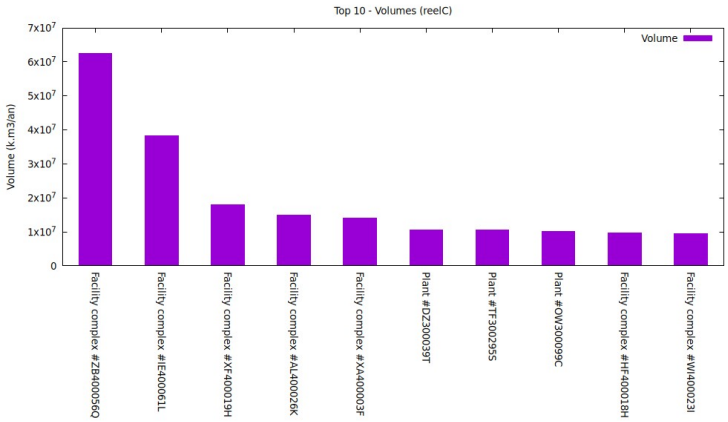
Axe Y : Volume d'eau ($k.m^3/an$).
Il permet d'identifier les usines qui fonctionnent à pleine capacité et celles qui sont sous-utilisées.

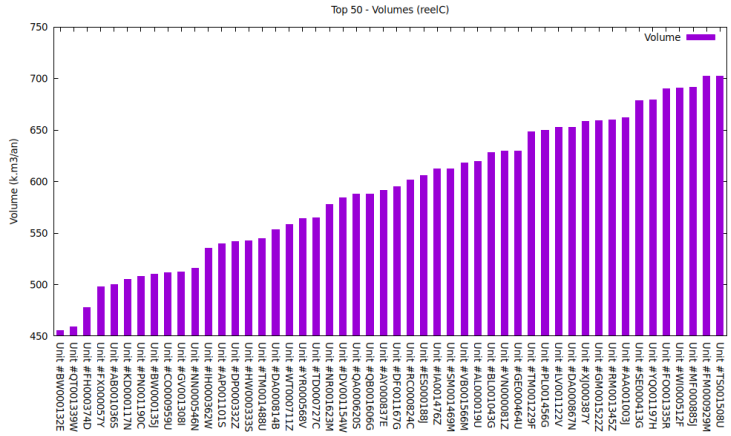
B. Histogramme des volumes totaux (captés et traités) par les sources.





Ces graphiques représentent les volumes d'eau stockés dans les sources usines.

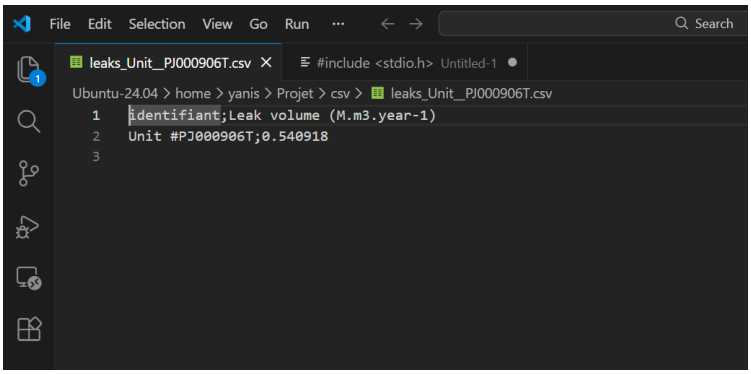




Ces graphiques représentent les volumes totaux réellement traités.

C. CSV des Pertes.

```
yanis@LAPTOP-MVDPK8GJ:~/Projet$ ./MyScript.sh c-wildwater_v3.csv leaks "Unit #PJ000906T"
Analyse des fuites pour l'usine : Unit #PJ000906T
Fichier de fuites généré : ./csv/leaks_Unit__PJ000906T.csv
--- Détails des fuites ---
identifiant;Leak volume (M.m3.year-1)
Unit #PJ000906T;0.540918
-e
Durée totale du script : 4871 ms
```



Exemple : Cette usine a eu une fuite d'eau de 0,54 M.m^3/an.

5. Conclusion

Ce projet nous a donc permis de comprendre l'importance d'une répartition des tâches flexible et d'une communication constante dans un projet comme celui-ci. L'utilisation des arbres AVL a été difficile pour traiter efficacement les données fournis.