

UNIVERSITÉ DE VERSAILLES
SAINT-QUENTIN-EN-YVELINES
DÉPARTEMENT DE INFORMATIQUE

MASTER 1 CALCUL HAUTE PERFORMANCE, SIMULATION

PPN- Profilage

Réalisé par :

- BOUCHELGA ABDELJALIL
- MOHAMED AITMHAND

Encadré par :

- DR. HUGO BOLLORÉ

Année Universitaire :2021-2022

TABLE DES MATIÈRES

List of figures	2
1 Option de compilation	3
1.1 Ajout d’une option de compilation dans le Makefile	3
Conclusion	4
Bibliographie	5

TABLE DES FIGURES

1.1 Ajout d'une option de compilation dans le Makefile

Jusqu'à présent on a optimiser un peu notre code, en modifiant sur le code lui même, il y a une autre astuce que nous pouvons implémenter sans changer sur le code, pour pousser encore notre optimisation. l'idée consiste à ajouter des indicateurs supplémentaire dans notre étape de compilation du code : *-Ofast -march=native -finline-functions -funroll-loops -ftree-vectorize* ces options permettent de maximiser l'optimisation.

- ***-Ofast*** : Est comme *-O3 -ffast-math* permet un niveau d'optimisation plus élevé et des optimisations plus agressives pour les calculs arithmétiques (comme la réassociation réelle).
- ***-funroll-loops*** : Permet le déroulement de la boucle.
- ***-finline-fonctions*** : Intègre toutes les fonctions simples dans les fonctions d'appel.
- ***-ftree-vectorize*** : Permet de vectoriser les boucles.
- ***-fopt-info-vecou* et *-fopt-info-vec-optimized*** Pour avoir des information sur la vectorisation.

Conclusion

Dans la première partie du projet, nous sommes partis d'un code exécuté sur un seul nœud et très coûteux en terme de performance et manque d'optimisation.

puis nous avons appliqué plusieurs pistes d'optimisation, comme le parallélisme, le réglage de la mémoire, l'auto vectorisation ..etc. pour faire réduire le temps d'exécution du code de 135.99812s d'origine à 11.211290s par itération. Il y a encore des opportunités d'optimisation qu'on peut apporter en parallélisation par MPI, le réglage de la vectorisation et d'autres choses qu'on peut le faire pour une future opportunité.

Finalement, ce projet nous a permis de mettre en pratique les techniques d'analyse et d'optimisation que nous avons vues au deuxième semestre, ainsi nous a fait connaître le but d'utilisation du calcul parallèle pour réduire le temps d'exécution et aussi fournir la possibilité de résoudre beaucoup de problèmes complexes dont la solution n'est pas possible lors d'utilisation d'un système informatique à processeur unique.

Dépôt Github : <https://github.com/Chaichas/Reconnaissance-de-Plantes-avec-NN>

BIBLIOGRAPHIE

- [1] "Profilier gprof", <https://ftp.traduc.org/doc-vf/gazette-linux/html/2004/100/lg100-L.html>
- [2] "Profilier perf", <https://linuxembedded.fr/2019/02/les-traceurs-sous-linux-22>
- [3] "Profilier gprof", <https://fr.wikipedia.org/wiki/Gprof>