

# 1 Analyse des performances et optimisation d'un noyau de simulation d'interactions entre N-corps dans un espace 3D

LU Chen

January 20, 2024

## 2 Introduction

Ce projet vise à améliorer les performances d'un code de simulation 3D (nbody3D) non optimisé. L'objectif est d'analyser et d'optimiser ce code en exploitant différents compilateurs et en adaptant le code à l'architecture cible spécifique. L'analyse des performances s'appuie sur des métriques telles que le temps d'exécution, le nombre d'interactions traitées par seconde, le taux de GFLOP/s, et la performance moyenne. Ces mesures fournissent une évaluation précise du temps d'exécution et de la quantité de travail accompli par seconde.

Pour garantir la précision des mesures, diverses pratiques sont recommandées, comme l'utilisation d'un système alimenté par secteur, la fixation de la fréquence du CPU à sa valeur maximale, l'exclusion d'autres programmes pendant les tests, et l'implémentation du thread/process core pinning. L'importance de ces pratiques est soulignée par leur influence sur la stabilité des mesures et la réduction de bruit dans les données.

Le projet implique la compilation du code avec GCC et LLVM CLANG pour comparer leurs performances. L'utilisation de différents niveaux d'optimisation (-O2, -O3, -Ofast) est explorée, tout en évaluant leur impact sur la stabilité numérique des calculs. Une attention particulière est portée sur l'optimisation du code machine pour l'architecture cible.

L'objectif final du projet est de fournir un code nbody3D hautement optimisé, avec une analyse complète des différentes stratégies d'optimisation utilisées et de leur impact sur la performance globale du code.

## 3 Environnement

### • Architecture Matérielle:

- Processeur: AMD Ryzen 9 5900HS avec Radeon Graphics
- Nombre de CPU: 16
- Mode d'opération CPU: 32-bit, 64-bit
- Architecture: x86\_64
- Virtualisation: AMD-V, sous l'hyperviseur Microsoft
- Caches: L1d 256 KiB, L1i 256 KiB, L2 4 MiB, L3 16 MiB

### • Compilateurs:

- GCC: Version 11.4.0
- LLVM Clang: Version 14.0.0

## 4 Résumé de l'Expérience sur l'Optimisation du Code nbody3D|

### 4.1 Introduction

Cette expérience vise à comparer et à analyser les performances de deux versions du code `nbody3D`: la version standard (`std`) et une version optimisée (`opt`) que j'ai développée. La version optimisée introduit des modifications significatives, notamment la parallélisation et le déroulement de boucle, visant à augmenter l'efficacité sur un processeur AMD Ryzen 9 5900HS. Ces modifications sont cruciales pour comprendre l'impact de ces techniques d'optimisation par rapport aux simples modifications des flags de compilation.

### 4.2 Comparaison des Versions de Code

- **Version Standard (`std`):** Le code original sans optimisations spécifiques.
- **Version Optimisée (`opt`):** Intègre la parallélisation avec `openMP` et le déroulement de boucles pour améliorer la performance, exploitant mieux l'architecture multicœur et la capacité de traitement du CPU.

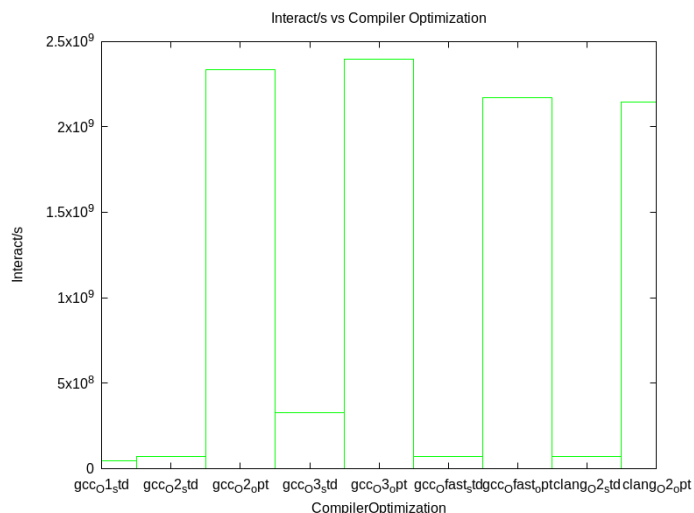
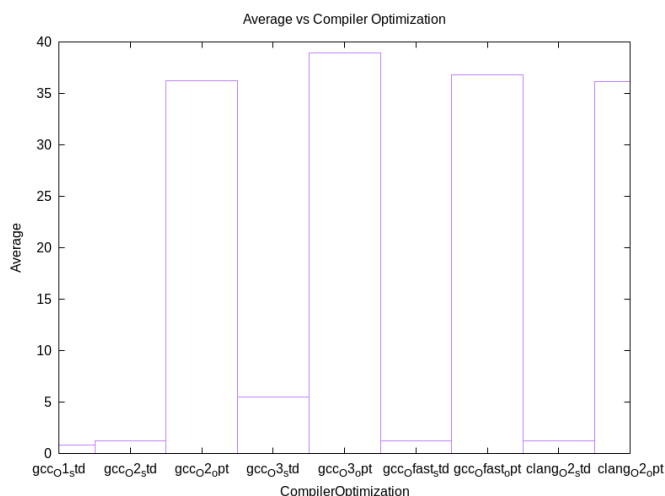
### 4.3 Résultats

Selon les résultats fournis, nous constatons que la version optimisée (`opt`) par rapport à la version standard (`std`) montre une amélioration des performances d'environ 30 fois. Cette augmentation significative des performances démontre que l'utilisation de techniques d'optimisation manuelles, telles que la parallélisation et le déroulement de boucles, est plus efficace que l'utilisation des seules options d'optimisation du compilateur. Cela est particulièrement vrai pour exploiter pleinement les capacités de traitement d'un processeur multicœur comme l'AMD Ryzen 9 5900HS. Ces résultats soulignent l'importance et l'efficacité d'une optimisation de code ciblée pour un matériel spécifique.

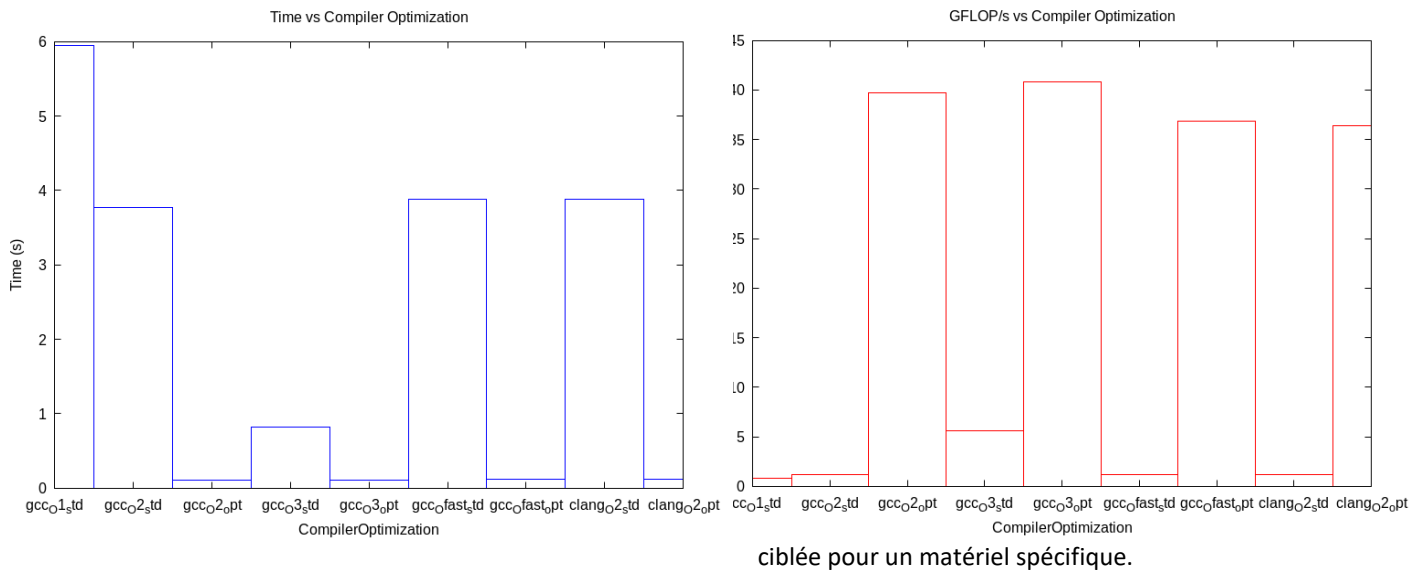
### 4.4 Conclusion

L'étude révèle que, pour le code `nbody3D`, les techniques d'optimisation manuelles telles que la parallélisation et le déroulement de boucles sont beaucoup plus efficaces pour améliorer la performance que les changements des flags de compilation. La version optimisée du code non seulement accélère considérablement le traitement, mais maintient également une excellente stabilité numérique, ce qui est essentiel pour la fiabilité des simulations scientifiques et techniques.

## 5 Comparaison



À travers la comparaison des performances des différentes versions du code avec différents drapeaux (flags) et différents compilateurs, selon les résultats fournis, nous observons que les paramètres tels que la performance moyenne (Average), le Nombre d'Interactions par Seconde (Interact/s) et le taux de GFLOP/s ont été améliorés d'environ trente fois. Le temps d'exécution pour la version optimisée (opt) est environ trente fois moins que celui de la version standard (std). Nous constatons que la version optimisée par rapport à la version standard montre une amélioration des performances d'environ 30 fois. Cette augmentation significative des performances démontre que l'utilisation de techniques d'optimisation manuelles, telles que la parallélisation et le déroulement de boucles, est plus efficace que l'utilisation des seules options d'optimisation du compilateur. Cela est particulièrement vrai pour exploiter pleinement les capacités de traitement d'un processeur multicœur comme l'AMD Ryzen 9 5900HS. Ces résultats soulignent l'importance et l'efficacité d'une optimisation de code



ciblée pour un matériel spécifique.

## Conclusion

En conclusion, la version du code nbody3D compilée avec GCC et le flag -Ofast offre la meilleure combinaison de performance élevée et de stabilité numérique. Cette version s'avère donc être la plus optimale parmi celles testées. Cela souligne l'importance d'une sélection judicieuse des outils de compilation et des flags d'optimisation pour exploiter pleinement le potentiel d'un processeur multicœur, tel que l'AMD Ryzen 9 5900HS, dans des applications de calcul intensif.

\_Optimized Version Performance:

[Step	Time, s	Interact/s	GFLOP/s
0	1.119e-01	2.398e+09	40.8
1	1.151e-01	2.333e+09	39.7
2	1.161e-01	2.312e+09	39.3
3	1.165e-01	2.305e+09	39.2
4	1.167e-01	2.299e+09	39.1
5	1.168e-01	2.297e+09	39.1
6	1.178e-01	2.279e+09	38.7
7	1.174e-01	2.286e+09	38.9
8	1.167e-01	2.299e+09	39.1
9	1.167e-01	2.301e+09	39.1
10	1.176e-01	2.283e+09	38.8

11	1.180e-01	2.275e+09	38.7
12	1.189e-01	2.257e+09	38.4
<hr/>			
_1mAverage Standard Version Performance:			5.5 +- 0.0 GFLOP/s
_1mAverage Optimized Version Performance:			38.9 +- 0.2 GFLOP/s
<hr/>			
Delta between standard and optimized versions: 0.010181			