Université de Versailles Saint Quentin en Yvelines

Master 1 :

Calcul Haute Performance, Simulation

RAPPORT SUR TRAVAIL D'OUTILS

BASES HPC

Nom et Prénom:

Chabane Khaled

SUR CE RAPPORT :

ABSTRACT :

Ce rapport présente une étude réalisée dans le cadre du programme de Master 1 en Calcul Haute Performance et Simulation à l'Université de Versailles Saint Quentin en Yvelines. L'objectif de ce travail est la mesure des performances de différentes versions de fonctions, telles que la multiplication de matrices, le produit scalaire et la réduction de tableaux, implémentées en langage C. L'étude explore l'impact de différents compilateurs, options d'optimisation et du déroulage de boucles sur les performances de ces fonctions.
Le rapport commence par assurer la stabilité de la machine, notamment en connectant l'ordinateur portable à une source d'alimentation, en stabilisant la fréquence du CPU et en épinglant le processus à un cœur spécifique.
Deux fonctions, dotprod et reduc, sont analysées avec une taille de matrice/vecteur de n=60, 100 répétitions. Sept indicateurs d'optimisation (-O0, -O1, -O3, -Ofast, -Og, -Os) et deux compilateurs (GCC et CLANG) sont utilisés.
Pour dotprod, trois versions (unroll8, unroll4 et CBLAS) sont ajoutées, et leurs performances sont évaluées avec différents niveaux d'optimisation. Les résultats montrent des variations en fonction des versions, des indicateurs d'optimisation et des compilateurs.

INTRODUCTION :

Ce travail repose sur la programmation en langage C pour prendre les mesures de performances de différentes versions de certaines fonctions, comme la multiplication de matrices, le produit scalaire et la réduction de tableaux. Dans ce travail on a collecté des mesures de performances et étudie l'impact des compilateurs et des options d'optimisation et du déroulage sur la performance de ces fonctions.

Dans ce rapport, je vais détailler les modifications que j'ai apportées aux codes sources, présenter les résultats que j'ai obtenus, et discuter des performances optimales que j'ai pu atteindre. Une partie essentielle de cette étude concerne l'analyse de l'impact du choix du compilateur et des options d'optimisation sur l'efficacité des programmes informatiques.

1- Information machine :

Tout au début, avant de commencer à collecter les mesures de performances, j'ai d'abord assuré que certaines contraintes étaient respectées. Cela inclut le fait que l'ordinateur

portable est connecté au secteur, que le CPU tourne à une fréquence stable, et que le processus est épinglé sur un cœur de calcul spécifique. Pour assurer la stabilité de la fréquence du CPU, j'ai utilisé la commande `cpupower` avec l'option `--governor performance` (sudo cpupower frequency-set --governor performance). De plus, j'ai utilisé la commande `taskset` pour épingler tous les processus au cœur 0 (taskset -c 0 ./leprogram).
Toutes les informations sur l'architecture cible ont été extraites, avec les détails sur le CPU obtenus à l'aide des commandes `lscpu` (lscpu > fichier.txt) et `cat /proc/cpuinfo` (cat /proc/cpuinfo > fichier.txt). Et les informations sur les données de caches aussi sont extraits en consultant les chemins : /sys/devices/system/cpu/cpu0/cache/index0/* pour L1 et /sys/devices/system/cpu/cpu0/cache/index2/* pour L2 et /sys/devices/system/cpu/cpu0/cache/index3/* pour L3.

2- Les fonctions :

-Dans le travail il y a trois différentes fonctions: dgemm, dotprod, reduc, nous allons parler dans ce rapport sur seulement dotprod et reduc.

-Spécifications :

-Pour la taille de matrices ou vecteurs on a utilisé n=60.
-Pour le nombre de répititions on a utilisé r=100.
-Pour le standard deviation (stddev) on a accepter les moins de 5 %.
- 7 flags d'optimisation sont utilisés (-O0, -O1, -O3, -Ofast, -Og, -Os).
- 2 compilateurs sont utilisés (GCC et CLANG).

2-1- dotprod :

La fonction dotprod implémente le calcul simple et direct du produit scalaire de deux vecteurs.
Premièrement j'ai compilé le programme dotprod avec "make CC=gcc OFLAGS=-O3" (le cmpilateur gcc et l'option d'optimisation -O3), après j'ai lancer le programme avec "taskset -c 0 ./dotprod 60 100" (la taille de matrices 60 et le nombre d'itération 100 et j'ai récolter les résultats et les a mis dans un fichier « .txt »

2-1-1-Modification du Makefile :

J'ai modifier le Makefile fournit afin de tester plusieurs flags d'optimisation et compilateurs. Je l'ai modifie de tel sorte qu'on compile avec la commande (make CC='The compilator' OFLAGS='The flag').

2-1-2-L'ajout de versions :

J'ai ajouté trois versions de dotprod la version unroll8 et unroll4 et cblas.

Unroll8 : Une fonction avec un déroulage de boucle x8, la boucle principale traite huit éléments à la fois. Cela signifie que, au lieu de faire une opération par itération de la boucle, huit opérations sont effectuées simultanément.

Unroll4 : Une fonction avec un déroulage de boucle x4, la boucle principale traite quatre éléments à la fois.

Cblas : (C Basic Linear Algebra Subprograms) est une interface standardisée pour les bibliothèques de sous-routines de base en algèbre linéaire. Cela inclut des opérations courantes telles que le produit matrice-vecteur, le produit matrice-matrice, les opérations de vecteur (comme la norme euclidienne), et d'autres fonctions associées.

2-1-3- Collection des données de performance :

Après avoir compilé le programme avec make CC=’compilateur’ OFLAGS=’flag’
j’ai collecté les données de performance en éxécutant le programme dotprod avec "taskset -c 0 ./dotprod 60 100 > fichier.txt" (taskset pour pinner le processus sur le coeur 0), et stocké le résultat dans un fichier ‘.txt’.
Avec deux compilateurs et 7 flags d’optimisation, j’ai eu 14 fichier en tout, 7 fichier pour chaque compilateur, un fichier pour chaque flag.

2-1-4-Compilateur GCC :

Les données de performance collecter selon GCC seront présenté pour chaque flag par la suite :

2-1-4-1-flag -O0 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	1996.695	129.900	131.790	130.225	0.339 (0.261 %)	3515.187
unroll4	0.938	0.001	0.000	60	100	1572.219	62.420	64.960	62.771	0.430 (0.685 %)	7292.609
unroll8	0.938	0.001	0.000	60	100	4075.396	127.430	128.480	127.743	0.212 (0.166 %)	3583.473
CBLAS	0.938	0.001	0.000	60	100	2413.003	44.810	55.440	45.247	1.832 (4.048 %)	10116.934

Pour ce flag les meilleures performances sont obtenus avec la version CBLAS et la dernière est obtenus avec la version BASE.

2-1-4-2-flag -O1 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	1373.356	37.100	40.690	37.328	0.610 (1.635 %)	12263.418
unroll4	0.938	0.001	0.000	60	100	3691.876	18.080	18.730	18.174	0.112 (0.617 %)	25187.497
unroll8	0.938	0.001	0.000	60	100	3621.982	36.500	37.070	36.605	0.099 (0.270 %)	12505.651
CBLAS	0.938	0.001	0.000	60	100	3355.091	40.960	53.180	41.468	2.103 (5.072 %)	11038.994

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-3-flag -O2 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	711.746	37.610	38.270	37.714	0.111 (0.294 %)	12137.882
unroll4	0.938	0.001	0.000	60	100	6435.236	19.690	20.750	19.779	0.179 (0.906 %)	23144.172
unroll8	0.938	0.001	0.000	60	100	3829.136	36.490	37.050	36.585	0.097 (0.266 %)	12512.384
CBLAS	0.938	0.001	0.000	60	100	2589.210	42.810	53.000	43.226	1.756 (4.062 %)	10589.993

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-4-flag -O3 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	2406.177	35.490	37.350	35.626	0.315 (0.883 %)	12849.124
unroll4	0.938	0.001	0.000	60	100	1012.902	29.240	33.260	29.427	0.689 (2.342 %)	15556.083
unroll8	0.938	0.001	0.000	60	100	18005.121	36.670	37.110	36.785	0.080 (0.218 %)	12444.251
CBLAS	0.938	0.001	0.000	60	100	1141.686	40.490	52.230	40.946	2.027 (4.949 %)	11179.593

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-5-flag -Ofast :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	818.601	18.820	20.530	19.049	0.273 (1.435 %)	24030.736
unroll4	0.938	0.001	0.000	60	100	1059.653	27.790	29.300	27.903	0.254 (0.909 %)	16405.696
unroll8	0.938	0.001	0.000	60	100	15978.877	39.120	40.910	39.317	0.297 (0.755 %)	11642.813
CBLAS	0.938	0.001	0.000	60	100	1705.888	40.180	53.350	40.692	2.274 (5.588 %)	11249.442

Pour ce flag les meilleures performances sont obtenus avec la version BASE et la dernière est obtenus avec la version CBLAS.

2-1-4-6-flag -Og :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	2240.644	37.200	37.990	37.358	0.135 (0.360 %)	12253.471
unroll4	0.938	0.001	0.000	60	100	1798.245	24.090	259.050	35.896	45.863 (127.767 %)	12752.477
unroll8	0.938	0.001	0.000	60	100	1995.353	37.530	38.280	37.864	0.107 (0.284 %)	12089.603
CBLAS	0.938	0.001	0.000	60	100	1458.803	40.890	55.870	41.492	2.582 (6.224 %)	11032.706

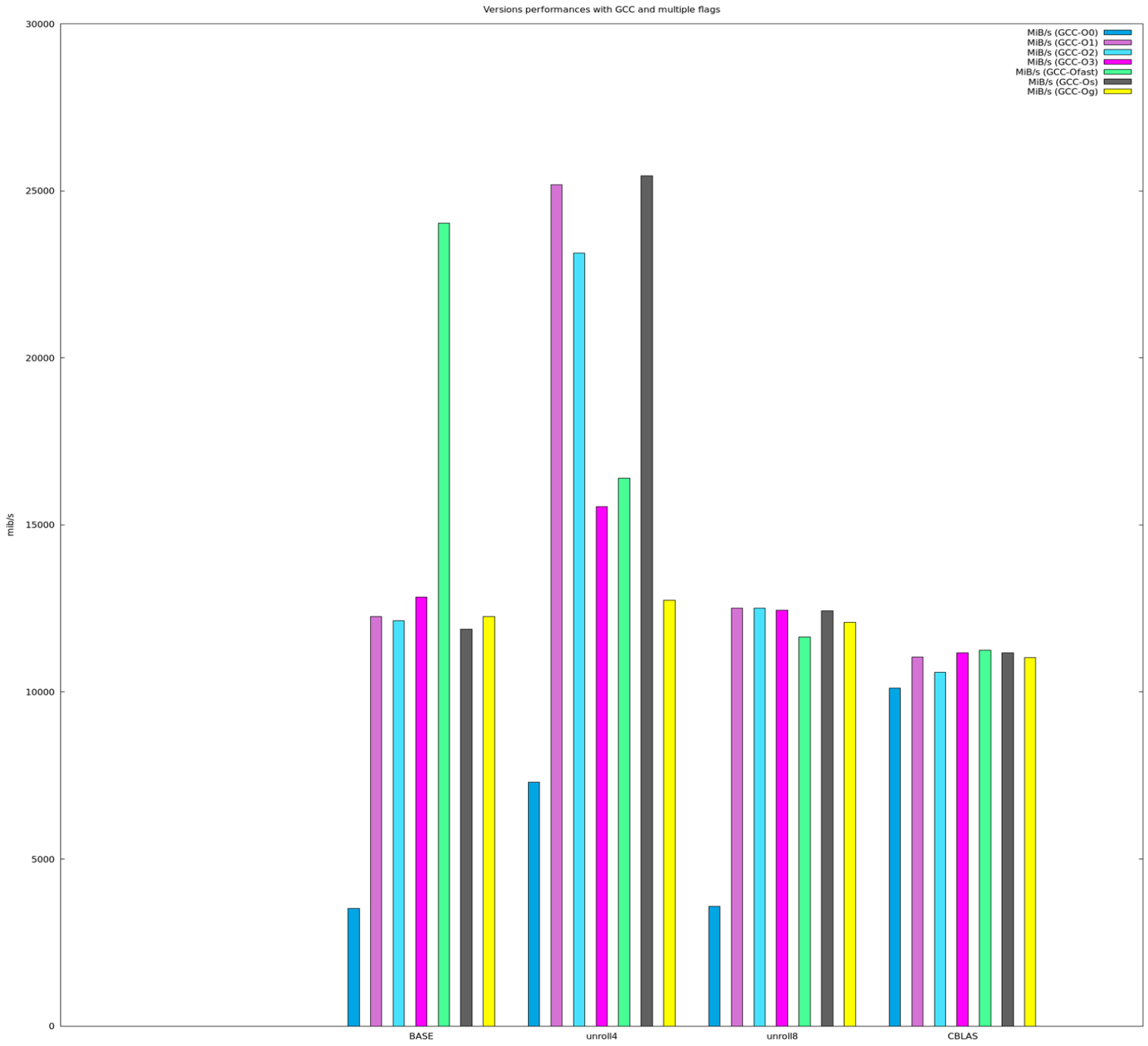
Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-7-flag -Os :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	1913.671	38.410	39.280	38.516	0.149 (0.386 %)	11885.008
unroll4	0.938	0.001	0.000	60	100	3666.163	17.910	18.650	17.988	0.123 (0.685 %)	25448.881
unroll8	0.938	0.001	0.000	60	100	1281.533	36.690	37.150	36.847	0.078 (0.211 %)	12423.476
CBLAS	0.938	0.001	0.000	60	100	1124.255	40.550	51.680	40.988	1.920 (4.685 %)	11168.269

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

Comparaison entre les versions de GCC et plusieurs flags :



Pour le compilateur GCC j'ai obtenus les meilleurs performances avec UNROLL4 avec le Flag -Os, et deuxièmement avec la même version mais le Flag -O3. Les dernière sont obtenus avec les versions UNROLL8 et BASE en Flag -O0, avec UNROLL8 est légèrement plus performant que BASE.

2-1-4-Compilateur CLANG :

Les données de performance collecter selon CLANG seront présenté pour chaque flag par la suite :

2-1-4-1-flag -O0 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	1393.826	135.030	137.320	135.327	0.390 (0.288 %)	3382.657
unroll4	0.938	0.001	0.000	60	100	1383.298	62.380	63.060	62.690	0.162 (0.258 %)	7302.021
unroll8	0.938	0.001	0.000	60	100	1252.610	126.710	128.510	127.114	0.297 (0.233 %)	3601.207
CBLAS	0.938	0.001	0.000	60	100	5312.749	41.750	55.130	42.251	2.313 (5.474 %)	10834.332

Pour ce flag les meilleures performances sont obtenus avec la version CBLAS et la dernière est obtenus avec la version BASE.

2-1-4-2-flag -O1 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	5676.405	36.410	37.250	36.558	0.138 (0.377 %)	12521.407
unroll4	0.938	0.001	0.000	60	100	5164.002	17.760	18.720	17.856	0.160 (0.895 %)	25636.319
unroll8	0.938	0.001	0.000	60	100	1463.626	36.010	311.000	44.471	47.847 (107.591 %)	10293.553
CBLAS	0.938	0.001	0.000	60	100	7394.790	40.750	53.060	41.235	2.124 (5.150 %)	11101.460

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version UNROLL8.

2-1-4-3-flag -O2 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	1459.447	36.190	340.610	50.594	57.012 (112.685 %)	9047.797
unroll4	0.938	0.001	0.000	60	100	1091.501	17.710	18.410	17.781	0.119 (0.671 %)	25744.672
unroll8	0.938	0.001	0.000	60	100	3223.724	36.180	36.610	36.271	0.077 (0.213 %)	12620.578
CBLAS	0.938	0.001	0.000	60	100	1156.662	40.510	54.660	41.045	2.445 (5.957 %)	11152.850

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version BASE.

2-1-4-4-flag -O3 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	656.159	36.210	37.950	36.388	0.287 (0.789 %)	12580.219
unroll4	0.938	0.001	0.000	60	100	2603.646	17.720	18.430	17.789	0.121 (0.683 %)	25732.831
unroll8	0.938	0.001	0.000	60	100	5283.680	36.050	36.480	36.231	0.092 (0.253 %)	12634.512
CBLAS	0.938	0.001	0.000	60	100	1011.294	40.240	51.100	40.658	1.876 (4.613 %)	11258.917

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-5-flag -Ofast :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	5822.016	11.560	12.970	11.654	0.238 (2.046 %)	39280.758
unroll4	0.938	0.001	0.000	60	100	774.544	18.000	19.480	18.104	0.250 (1.380 %)	25285.731
unroll8	0.938	0.001	0.000	60	100	4497.425	17.650	387.390	30.043	64.234 (213.808 %)	15236.934
CBLAS	0.938	0.001	0.000	60	100	2391.988	40.330	176.870	45.188	23.849 (52.777 %)	10130.164

Pour ce flag les meilleures performances sont obtenus avec la version BASE et la dernière est obtenus avec la version CBLAS.

2-1-4-6-flag -Og :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	3751.089	36.460	37.220	36.562	0.133 (0.364 %)	12520.058
unroll4	0.938	0.001	0.000	60	100	1048.574	17.770	18.540	17.848	0.130 (0.728 %)	25647.636
unroll8	0.938	0.001	0.000	60	100	1081.340	35.960	36.410	36.046	0.081 (0.224 %)	12699.409
CBLAS	0.938	0.001	0.000	60	100	1040.842	40.780	51.740	41.212	1.891 (4.588 %)	11107.419

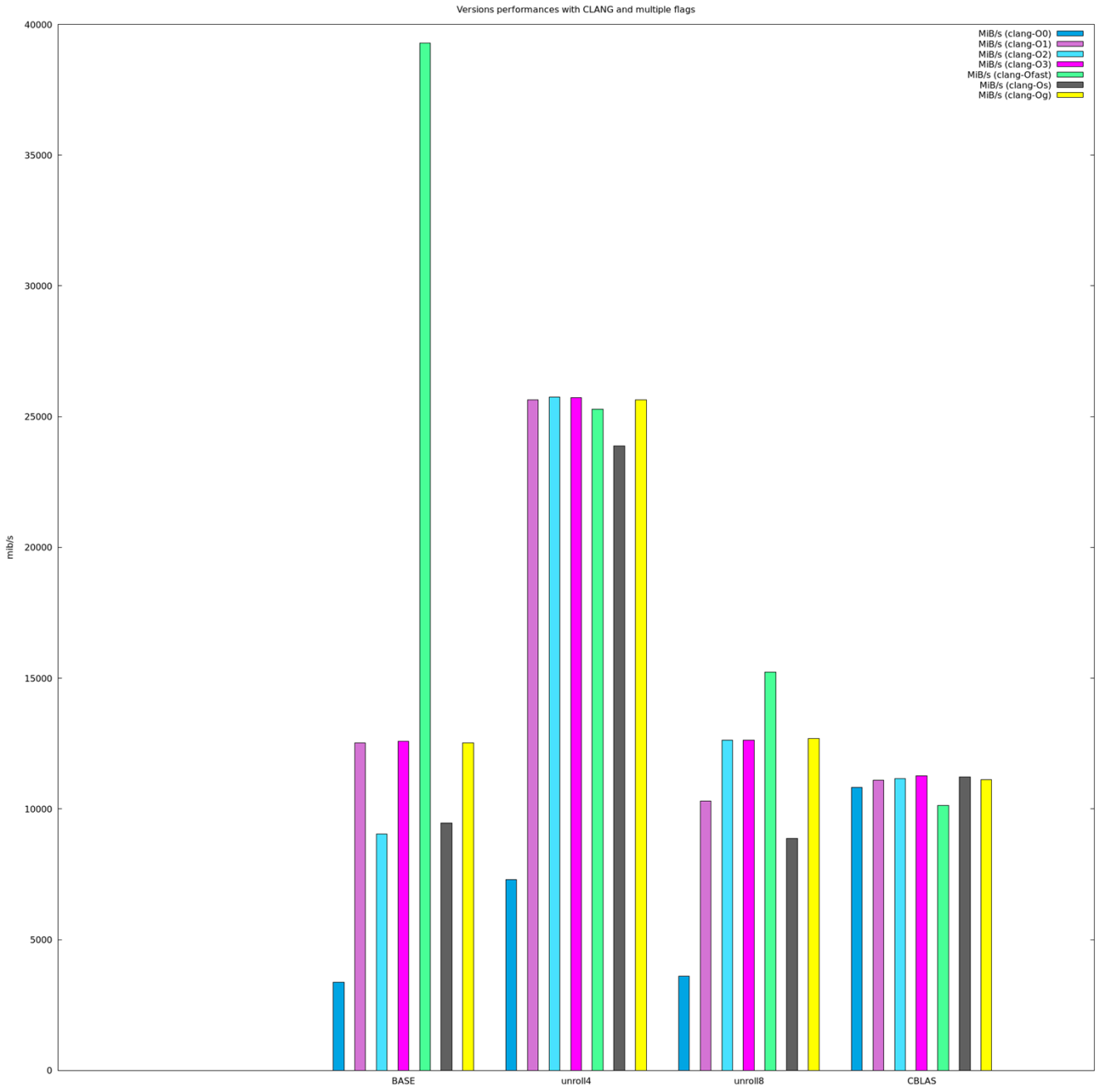
Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-7-flag -Os :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	1647.502	48.200	49.510	48.351	0.219 (0.454 %)	9467.530
unroll4	0.938	0.001	0.000	60	100	1635.382	19.070	19.920	19.171	0.146 (0.761 %)	23878.414
unroll8	0.938	0.001	0.000	60	100	3575.156	39.340	189.430	51.612	29.415 (56.992 %)	8869.253
CBLAS	0.938	0.001	0.000	60	100	6902.401	40.290	54.410	40.813	2.442 (5.984 %)	11216.199

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version UNROLL8.

Comparaison entre les versions de CLANG et plusieurs flags :

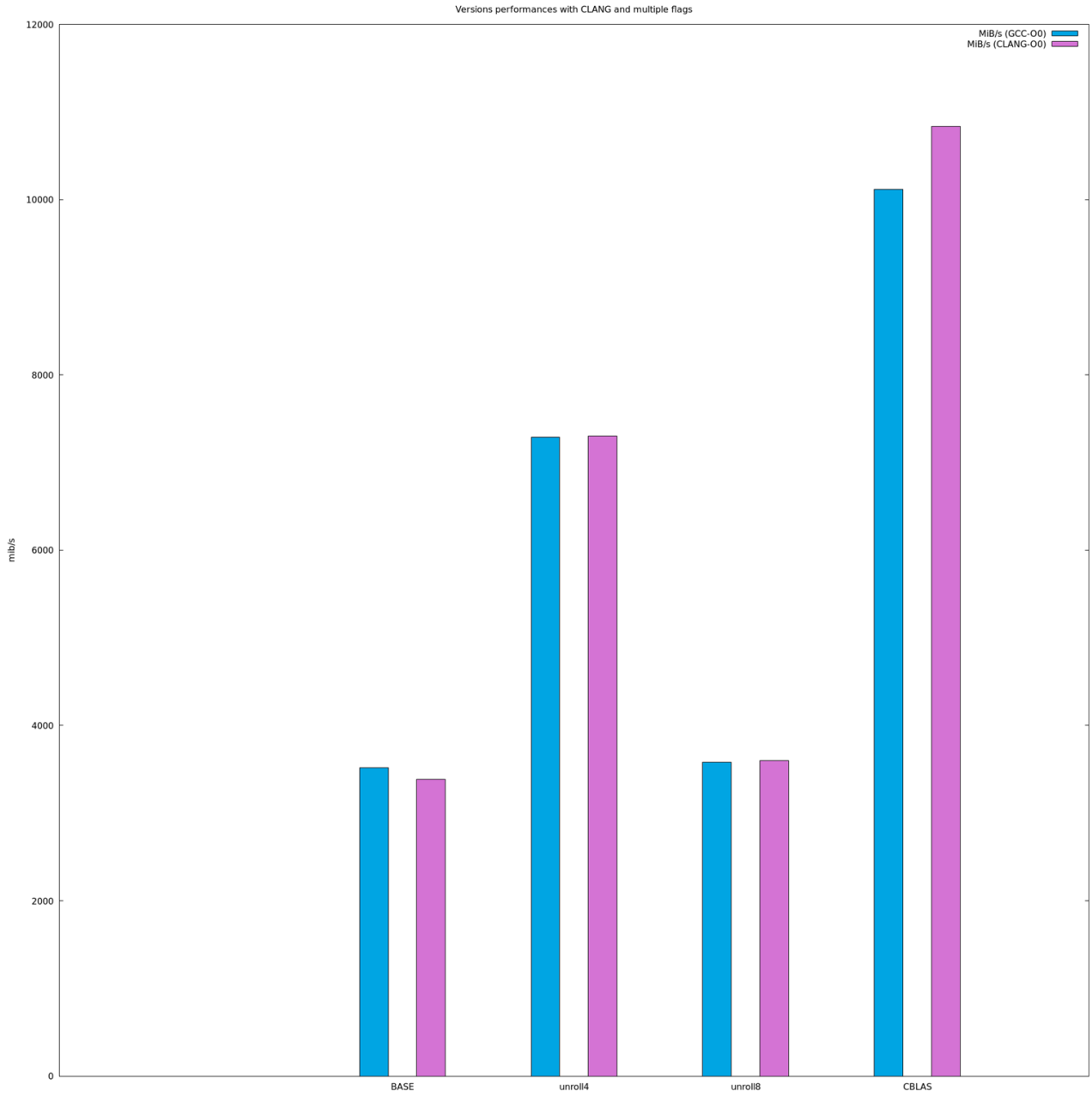


Pour le compilateur CLANG j'ai obtenus les meilleurs performances avec la version BASE une fois avec le flag -Ofast et presque pour tout les autres flags UNROLL4 a les meilleurs performances. Les dernières sont obtenus avec les versions UNROLL8 et BASE avec le Flag -O0, avec UNROLL8 est légèrement plus performant que BASE.

Comparaison entre les version des deux compilateurs :

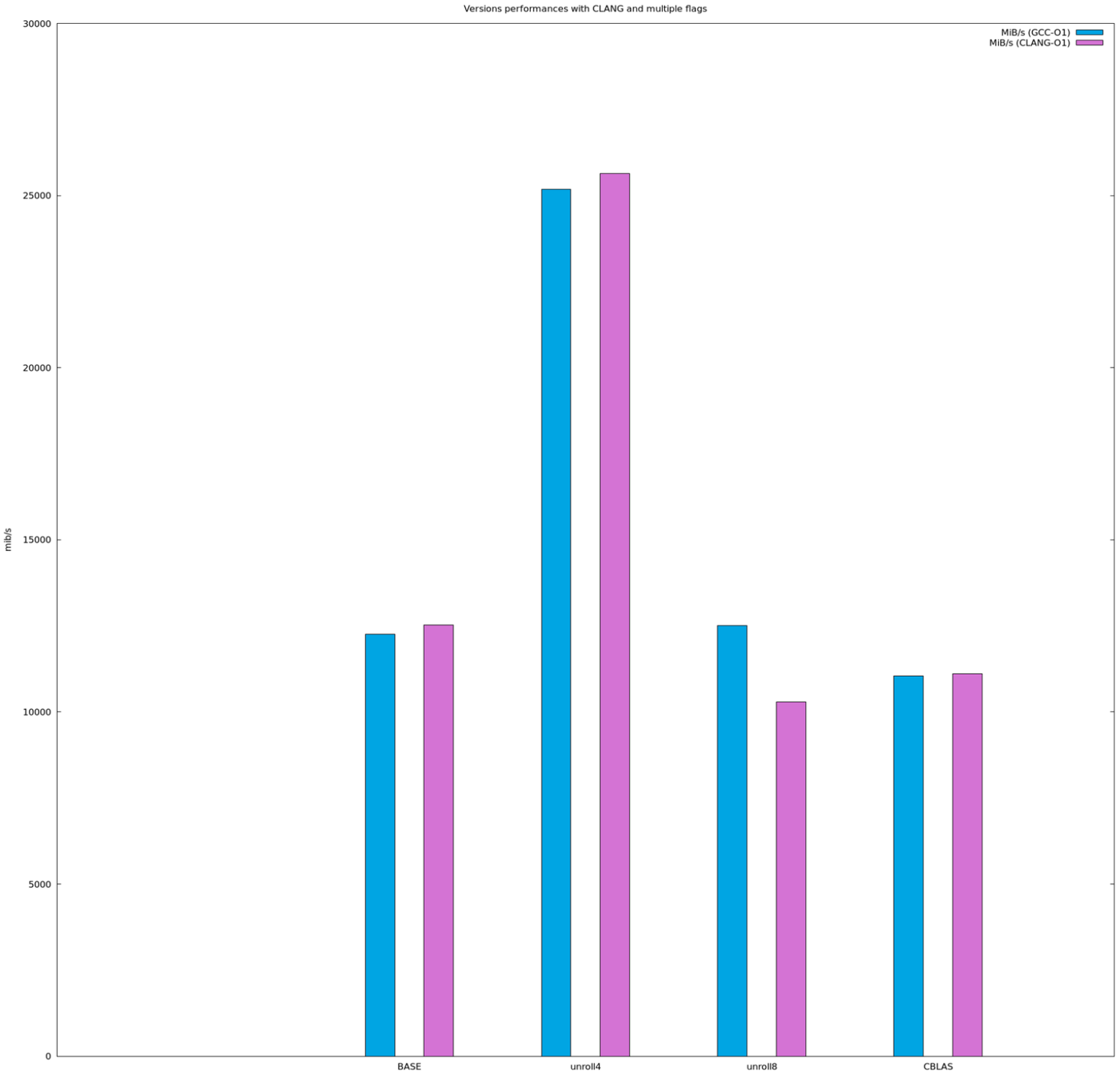
Comparaison selon chaque flag :

Flag-O0 :



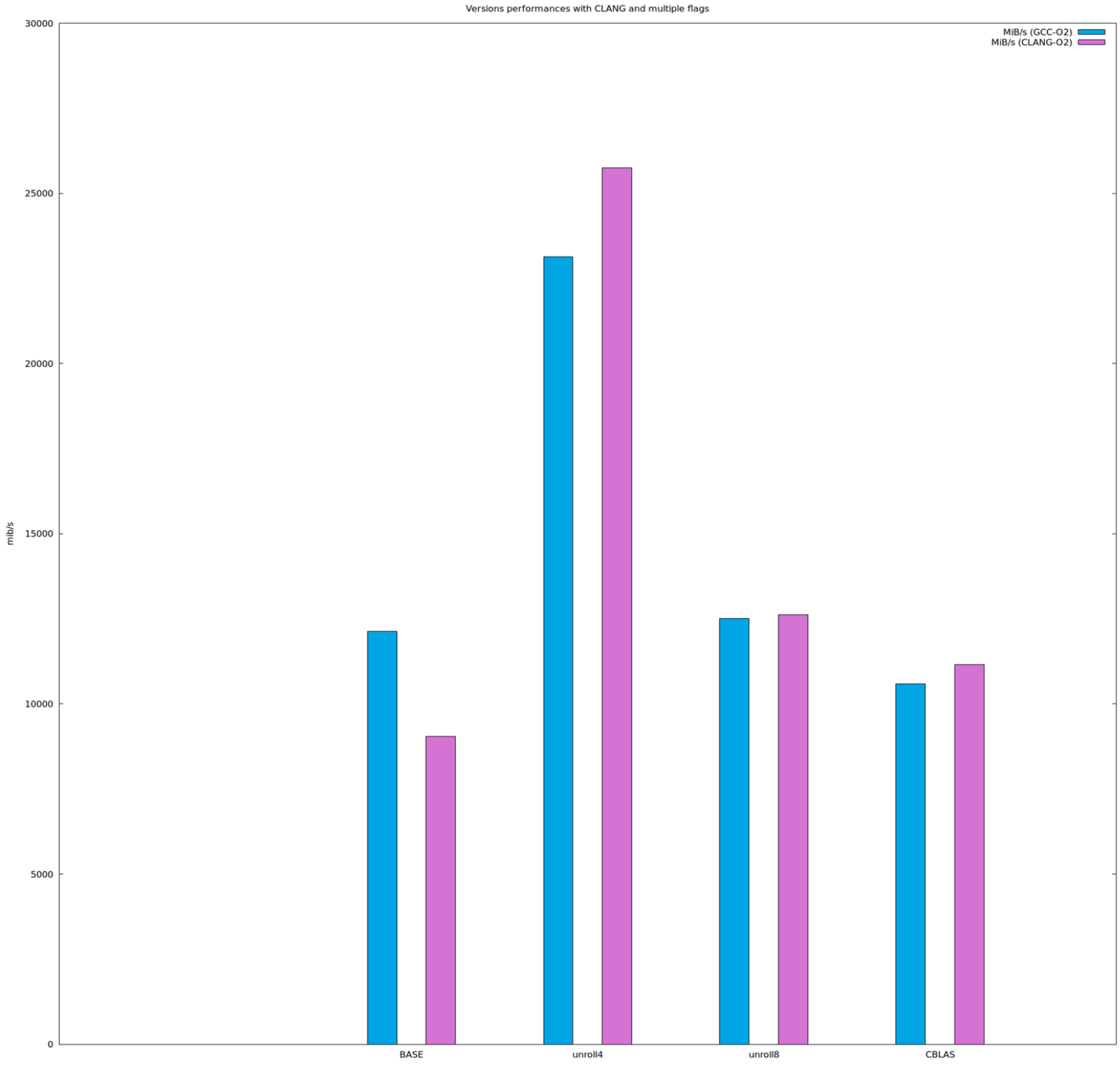
Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version CBLAS et la dernière est obtenus par CLANG avec la version BASE. Toutes les versions de CLANG dépassent les versions de GCC sauf la version BASE (BASE-GCC est mieux que BASE-CLANG).

Flag-O1 :



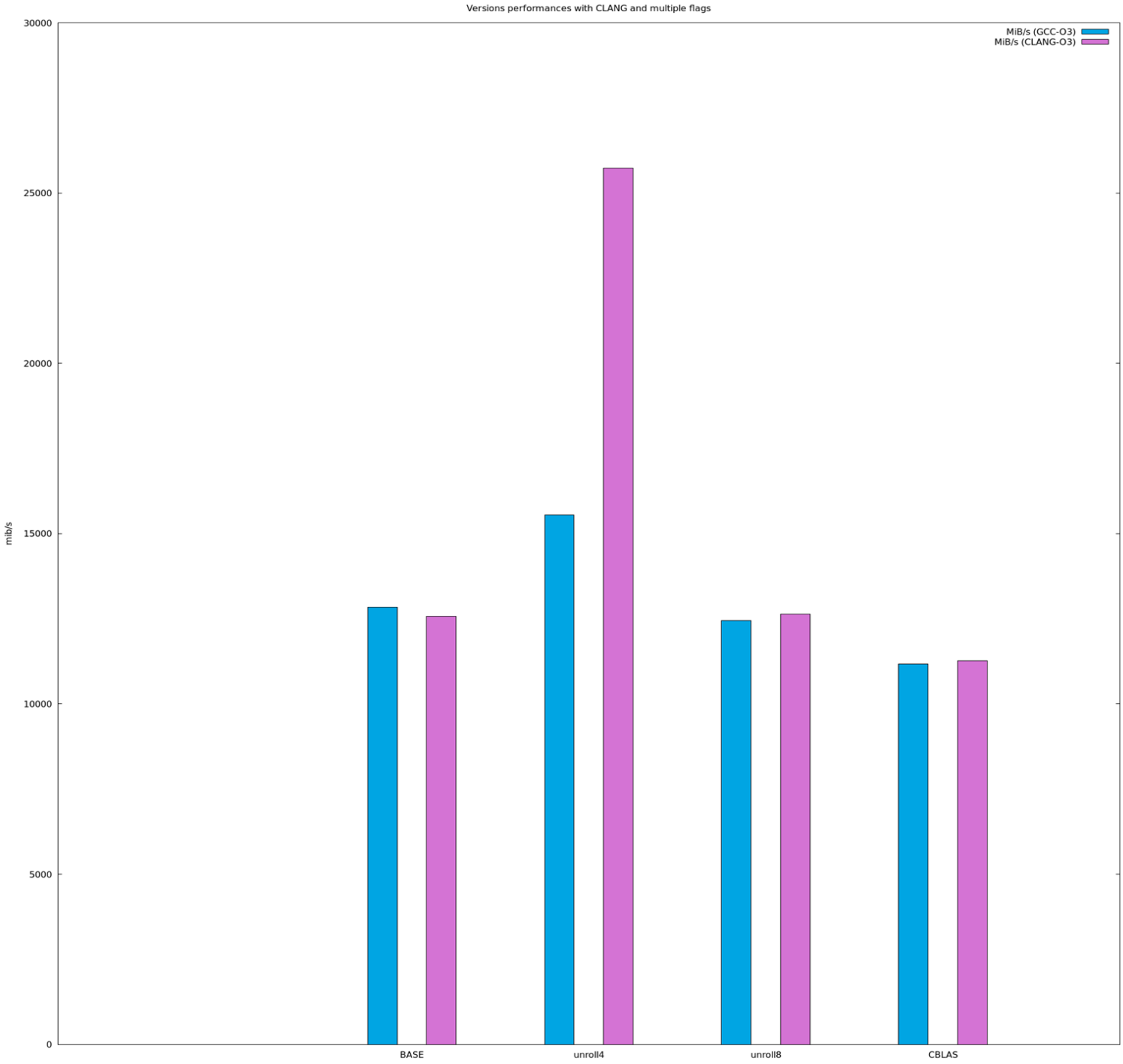
Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version UNROLL4 et la dernière est obtenus par CLANG avec la version UNROLL8. Toutes les versions de CLANG dépassent les versions de GCC sauf la version UNROLL8 (UNROLL8-GCC est mieux que UNROLL8-CLANG).

Flag-O2 :



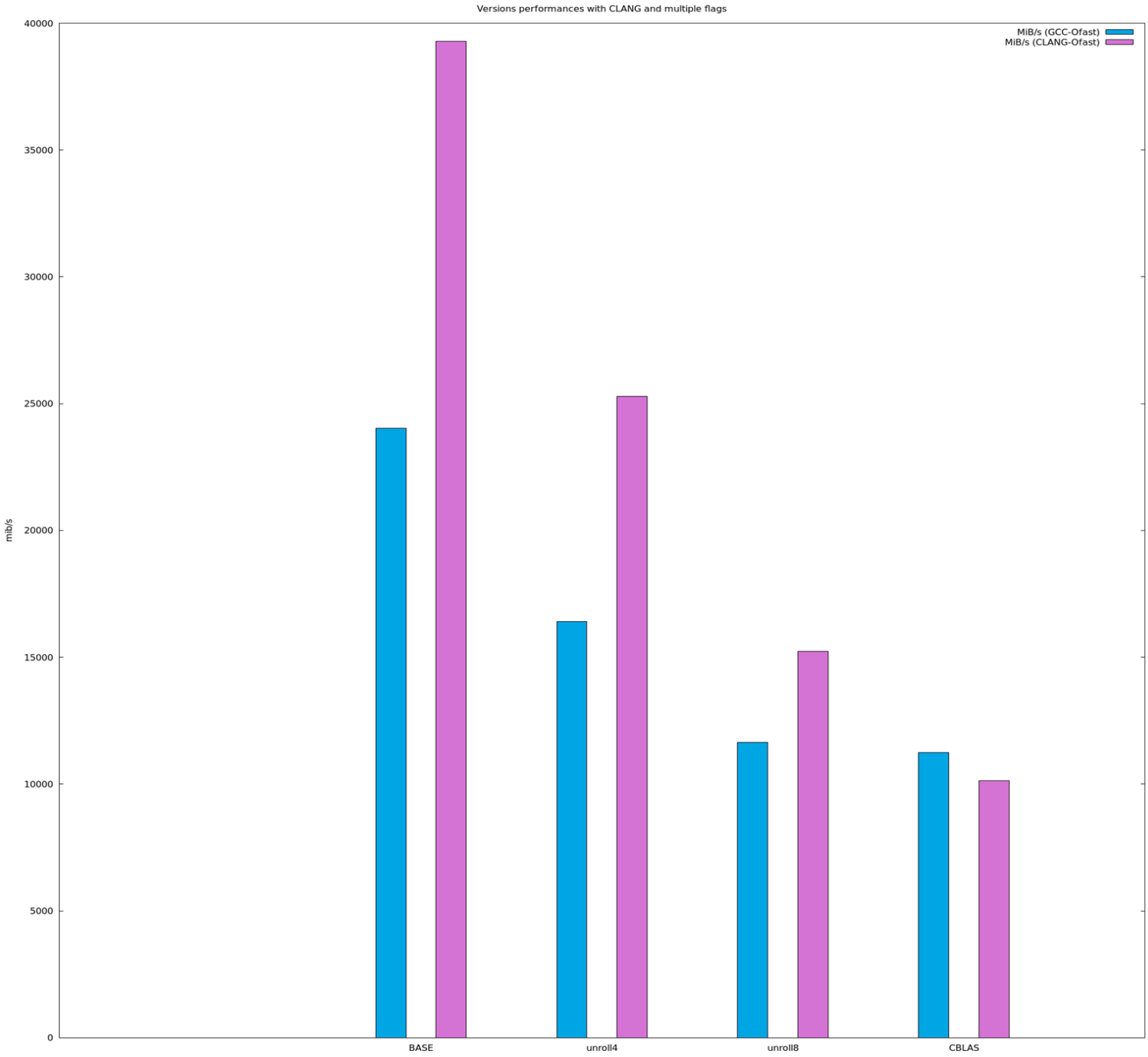
Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version UNROLL4 et la dernière est obtenus par CLANG avec la version BASE. Toutes les versions de CLANG dépassent les versions de GCC sauf la version BASE (BASE-GCC est mieux que BASE-CLANG).

Flag-O3 :



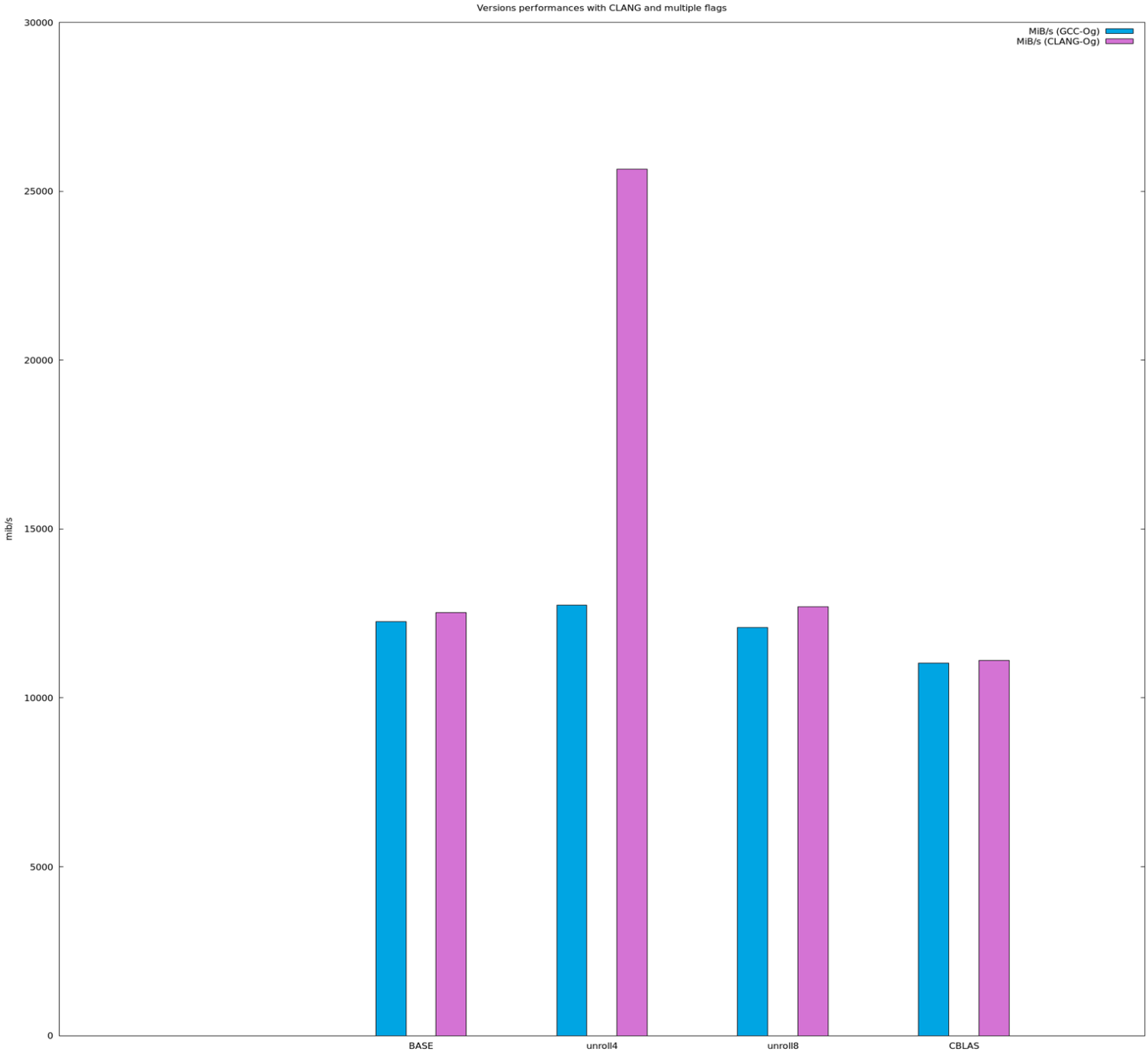
Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version UNROLL4 et la dernière est obtenus par CLANG avec la version BASE. Toutes les versions de CLANG dépassent les versions de GCC sauf la version BASE (BASE-GCC est mieux que BASE-CLANG).

Flag-Ofast :



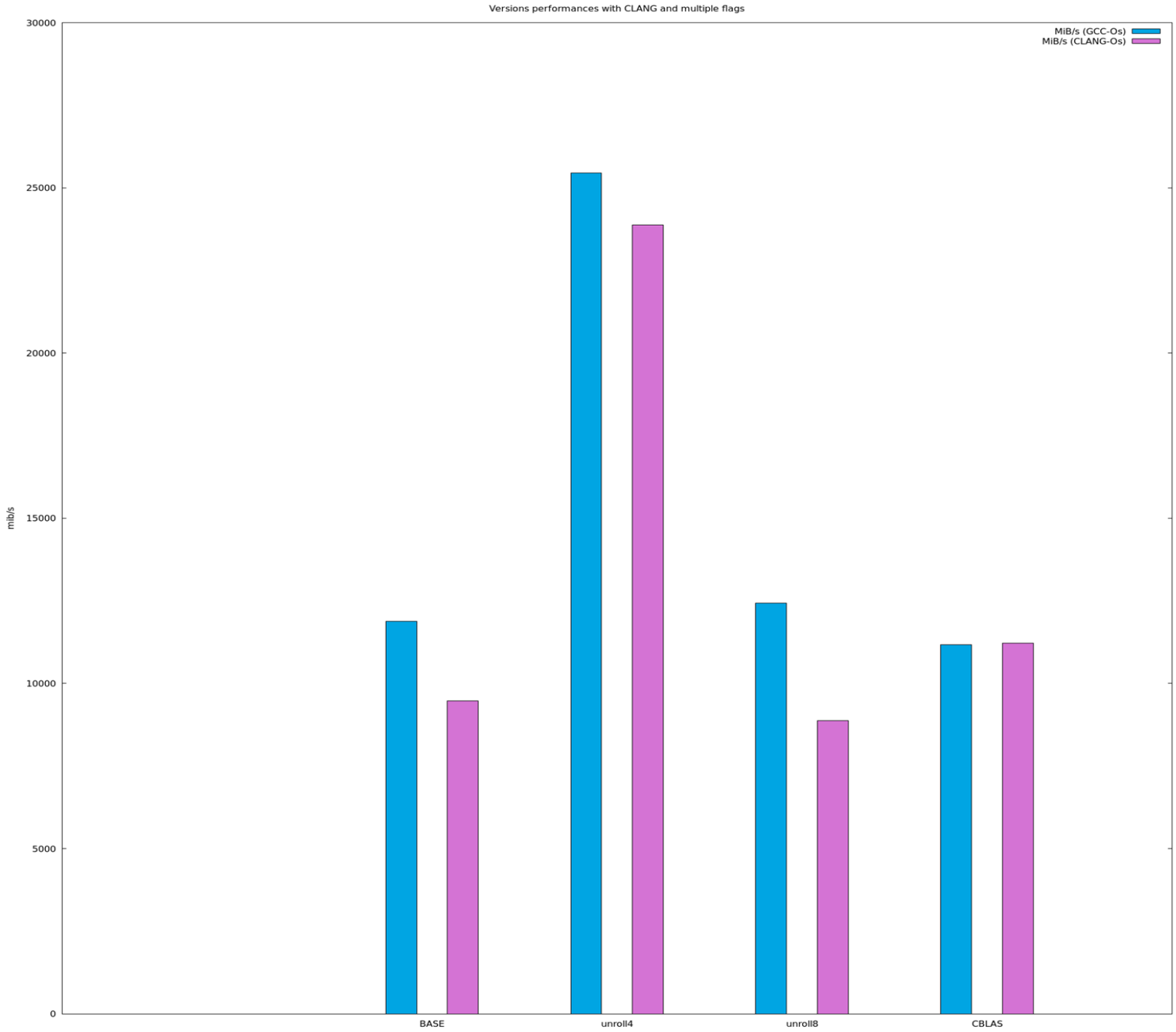
Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version BASE et la dernière est obtenus par CLANG avec la version CBLAS. Toutes les versions de CLANG dépassent les versions de GCC sauf la version CBLAS (CBLAS-GCC est mieux que CBLAS-CLANG).

Flag-Og :



Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version UNROLL4 et la dernière est obtenus par GCC avec la version CBLAS. Toutes les versions de CLANG dépassent les versions de GCC.

Flag-Os :



Pour ce flag les meilleures performances sont obtenus par le compilateur GCC avec la version UNROLL4 et la dernière est obtenus par CLANG avec la version UNROLL8. Toutes les versions de GCC dépassent les versions de CLANG sauf la version CBLAS (CBLAS-CLANG est mieux que CBLAS-GCC).

Conclusion sur dotprod :

Pour la fonction dotprod, des versions avec déroulage de boucle (unroll8 et unroll4) ainsi qu'une version utilisant CBLAS ont été ajoutées. Les performances ont été évaluées avec différents niveaux d'optimisation et deux compilateurs, GCC et CLANG. Les résultats ont montré des variations significatives en fonction des versions, des flags d'optimisation et des compilateurs utilisés.

2-2-reduc :

Cette fonction implémente une opération de réduction, en calculant la somme de tous les éléments d'un tableau a.
Premièrement j'ai compilé le programme reduc avec "make CC=gcc OFLAGS=-O3" (le cmpilateur gcc et l'option d'optimisation -O3), après j'ai lancer le programme avec "taskset -c 0 ./reduc 60 100" (la taille de matrices 60 et le nombre d'itération 100 et j'ai récolter les résultats et les a mis dans un fichier « .txt »

2-1-1-Modification du Makefile :

J'ai modifier le Makefile fournit afin de tester plusieurs flags d'optimisation et compilateurs. Je l'ai modifie de tel sorte qu'on compile avec la commande (make CC='The compilator' OFLAGS='The flag').

2-1-2-L'ajout de versions :

J'ai ajouté trois versions de reduc la version unroll8 et unroll4 et cblas.

2-1-3- Collection des données de performance :

Après avoir compilé le programme avec make CC='compilateur' OFLAGS='flag'
j'ai collecté les données de performance en éxécutant le programme reduc avec "taskset -c 0 ./reduc 60 100 > fichier.txt" (taskset pour pinner le processus sur le coeur 0), et stocké le résultat dans un fichier '.txt'.
Avec deux compilateurs et 7 flags d'optimisation, j'ai eu 14 fichier en tout, 7 fichier pour chaque compilateur, un fichier pour chaque flag.

2-1-4-Compilateur GCC :

2-1-4-1-Flag-O0 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	287.075	122.860	473.310	145.260	71.568 (49.269 %)	3151.334
BASE_UNROLL4	0.938	0.001	0.000	60	100	419.167	114.730	116.410	115.925	0.352 (0.303 %)	3948.776
BASE_UNROLL8	0.938	0.001	0.000	60	100	357.595	116.820	118.360	117.407	0.403 (0.343 %)	3898.958
CBLAS	0.938	0.001	0.000	60	100	639.965	43.580	55.240	44.068	2.009 (4.559 %)	10387.766

Pour ce flag les meilleures performances sont obtenus avec la version CBLAS et la dernière est obtenus avec la version BASE.

2-1-4-1-Flag-O1 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	668.779	32.000	33.210	32.106	0.203 (0.631 %)	14257.859
BASE_UNROLL4	0.938	0.001	0.000	60	100	473.975	28.750	29.170	28.821	0.076 (0.263 %)	15882.874
BASE_UNROLL8	0.938	0.001	0.000	60	100	287.804	28.750	29.260	28.823	0.088 (0.306 %)	15881.705
CBLAS	0.938	0.001	0.000	60	100	335.816	42.620	53.140	43.126	1.800 (4.174 %)	10614.549

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-O2 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	497.688	31.970	33.070	32.068	0.185 (0.577 %)	14274.970
BASE_UNROLL4	0.938	0.001	0.000	60	100	482.973	28.290	28.820	28.394	0.088 (0.310 %)	16122.052
BASE_UNROLL8	0.938	0.001	0.000	60	100	452.645	28.650	29.110	28.745	0.076 (0.265 %)	15924.900
CBLAS	0.938	0.001	0.000	60	100	474.692	42.170	54.760	42.647	2.176 (5.102 %)	10733.868

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-O3 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	328.151	32.080	33.660	32.242	0.261 (0.809 %)	14197.958
BASE_UNROLL4	0.938	0.001	0.000	60	100	356.349	30.540	31.750	30.748	0.205 (0.667 %)	14887.650
BASE_UNROLL8	0.938	0.001	0.000	60	100	245.572	28.570	28.750	28.655	0.049 (0.170 %)	15974.748
CBLAS	0.938	0.001	0.000	60	100	290.535	45.570	56.280	46.238	1.815 (3.926 %)	9900.058

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL8 et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-Ofast :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	526.060	14.070	15.140	14.152	0.179 (1.268 %)	32345.941
BASE_UNROLL4	0.938	0.001	0.000	60	100	559.297	14.820	15.980	14.895	0.196 (1.319 %)	30733.645
BASE_UNROLL8	0.938	0.001	0.000	60	100	234.723	18.360	19.330	18.461	0.162 (0.879 %)	24795.971
CBLAS	0.938	0.001	0.000	60	100	239.000	42.110	56.070	42.629	2.414 (5.662 %)	10738.293

Pour ce flag les meilleures performances sont obtenus avec la version BASE et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-Og :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	360.689	32.120	33.480	32.295	0.220 (0.682 %)	14174.644
BASE_UNROLL4	0.938	0.001	0.000	60	100	222.392	28.370	28.720	28.450	0.063 (0.221 %)	16090.111
BASE_UNROLL8	0.938	0.001	0.000	60	100	234.876	28.740	29.260	28.827	0.090 (0.314 %)	15879.535
CBLAS	0.938	0.001	0.000	60	100	416.760	42.140	55.140	42.651	2.243 (5.260 %)	10732.800

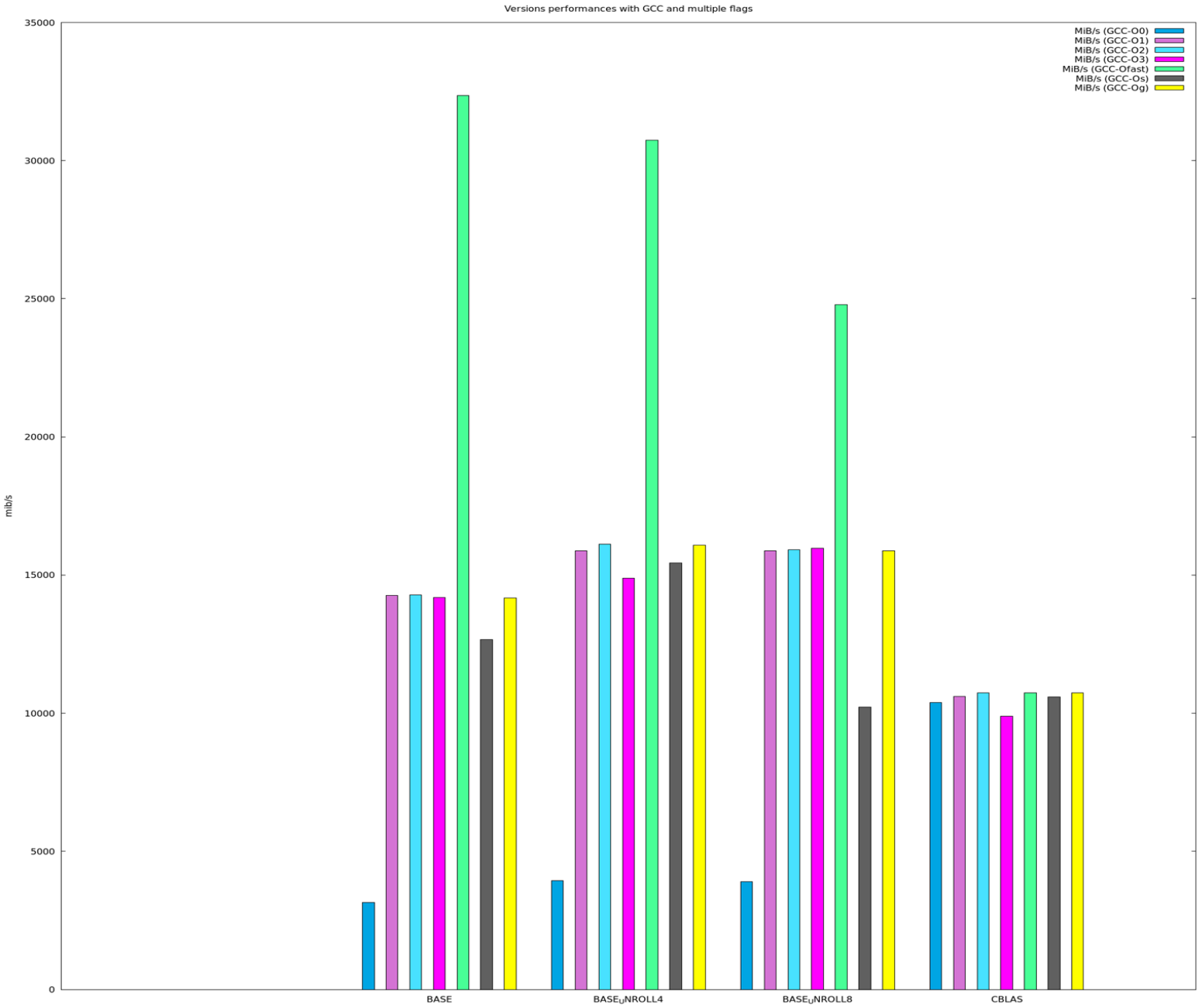
Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-O0s :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	223.975	35.980	37.340	36.128	0.229 (0.634 %)	12670.755
BASE_UNROLL4	0.938	0.001	0.000	60	100	360.094	29.550	29.950	29.643	0.069 (0.233 %)	15442.698
BASE_UNROLL8	0.938	0.001	0.000	60	100	230.895	29.560	264.900	44.766	48.813 (109.039 %)	10225.617
CBLAS	0.938	0.001	0.000	60	100	3786.306	42.770	54.360	43.222	2.001 (4.629 %)	10591.107

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version UNROLL8.

Comparaison entre les versions de GCC et plusieurs flags :



Pour le compilateur GCC j'ai obtenus les meilleurs performances avec la version BASE avec le flag -Ofast, toutes les version ont les meilleurs performances avec le flag -Ofast. La dernière performance est obtenus avec BASE avec le flag -O0, qui montre une performance faible aussi pour UNROLL8 et UNROLL4.

2-1-5-Compilateur CLANG :

2-1-4-1-Flag-O0 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	256.521	122.030	124.590	122.747	0.406 (0.331 %)	3729.328
BASE_UNROLL4	0.938	0.001	0.000	60	100	464.219	114.780	115.950	115.634	0.250 (0.216 %)	3958.731
BASE_UNROLL8	0.938	0.001	0.000	60	100	280.227	117.120	118.170	117.461	0.243 (0.207 %)	3897.168
CBLAS	0.938	0.001	0.000	60	100	400.731	42.390	54.550	42.903	2.093 (4.879 %)	10669.728

Pour ce flag les meilleures performances sont obtenus avec la version CBLAS et la dernière est obtenus avec la version BASE.

2-1-4-1-Flag-O1 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	1119.206	31.890	32.650	31.988	0.127 (0.398 %)	14310.264
BASE_UNROLL4	0.938	0.001	0.000	60	100	387.309	28.260	30.000	28.557	0.316 (1.105 %)	16029.670
BASE_UNROLL8	0.938	0.001	0.000	60	100	444.525	28.610	28.930	28.689	0.060 (0.210 %)	15956.187
CBLAS	0.938	0.001	0.000	60	100	199.056	41.880	54.100	42.499	2.087 (4.910 %)	10771.140

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-O2 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	330.857	28.630	29.430	28.723	0.134 (0.465 %)	15937.333
BASE_UNROLL4	0.938	0.001	0.000	60	100	168.546	28.760	29.180	28.895	0.083 (0.289 %)	15842.231
BASE_UNROLL8	0.938	0.001	0.000	60	100	476.879	28.700	29.210	28.782	0.089 (0.309 %)	15904.278
CBLAS	0.938	0.001	0.000	60	100	337.522	41.730	54.160	42.338	2.126 (5.020 %)	10812.077

Pour ce flag les meilleures performances sont obtenus avec la version BASE et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-O3 :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	481.487	28.630	29.850	28.729	0.205 (0.714 %)	15933.635
BASE_UNROLL4	0.938	0.001	0.000	60	100	481.852	28.660	29.210	28.881	0.092 (0.319 %)	15849.877
BASE_UNROLL8	0.938	0.001	0.000	60	100	316.081	28.620	29.280	28.722	0.115 (0.399 %)	15937.838
CBLAS	0.938	0.001	0.000	60	100	360.285	41.780	54.740	42.313	2.234 (5.280 %)	10818.504

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL8 et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-Ofast :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	1198.003	7.160	8.510	7.227	0.232 (3.208 %)	63341.026
BASE_UNROLL4	0.938	0.001	0.000	60	100	2416.943	14.650	16.010	14.749	0.228 (1.548 %)	31036.100
BASE_UNROLL8	0.938	0.001	0.000	60	100	259.214	10.460	11.200	10.521	0.126 (1.199 %)	43509.897
CBLAS	0.938	0.001	0.000	60	100	57090.324	41.860	53.850	42.322	2.071 (4.893 %)	10816.257

Pour ce flag les meilleures performances sont obtenus avec la version BASE et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-Og :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	337.702	31.890	33.160	32.002	0.215 (0.671 %)	14304.167
BASE_UNROLL4	0.938	0.001	0.000	60	100	370.462	28.280	28.740	28.549	0.161 (0.564 %)	16034.094
BASE_UNROLL8	0.938	0.001	0.000	60	100	414.358	28.620	29.040	28.699	0.072 (0.252 %)	15950.459
CBLAS	0.938	0.001	0.000	60	100	548.249	41.920	53.780	42.451	2.037 (4.799 %)	10783.366

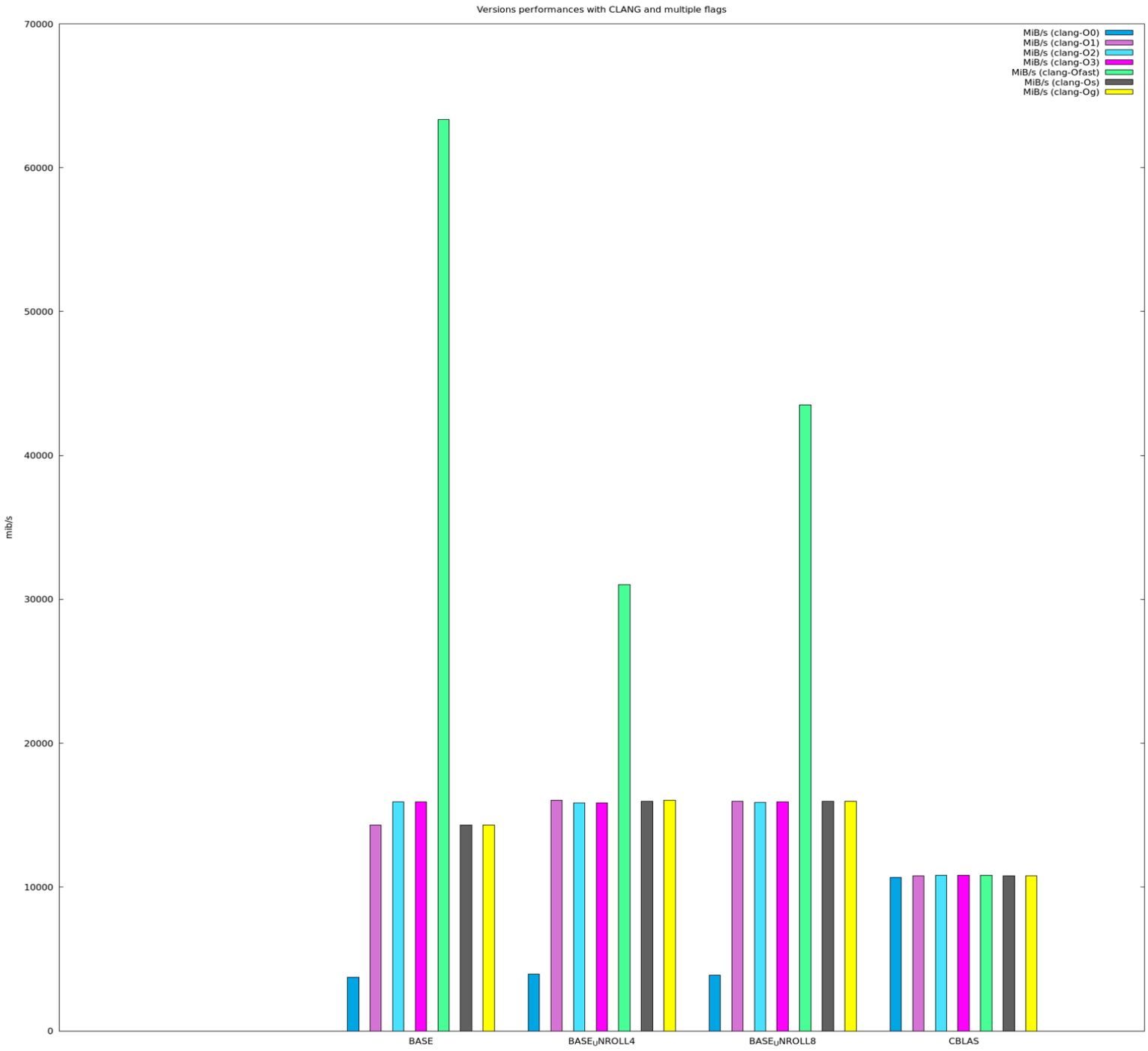
Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

2-1-4-1-Flag-Os :

title	KiB	MiB	GiB	n	r	d	min	max	mean	stddev (%)	MiB/s
BASE	0.938	0.001	0.000	60	100	253.962	31.890	33.340	32.008	0.244 (0.763 %)	14301.458
BASE_UNROLL4	0.938	0.001	0.000	60	100	697.338	28.450	28.830	28.649	0.068 (0.237 %)	15978.297
BASE_UNROLL8	0.938	0.001	0.000	60	100	361.908	28.600	29.040	28.679	0.078 (0.272 %)	15961.414
CBLAS	0.938	0.001	0.000	60	100	377.995	41.760	56.600	42.442	2.545 (5.995 %)	10785.752

Pour ce flag les meilleures performances sont obtenus avec la version UNROLL4 et la dernière est obtenus avec la version CBLAS.

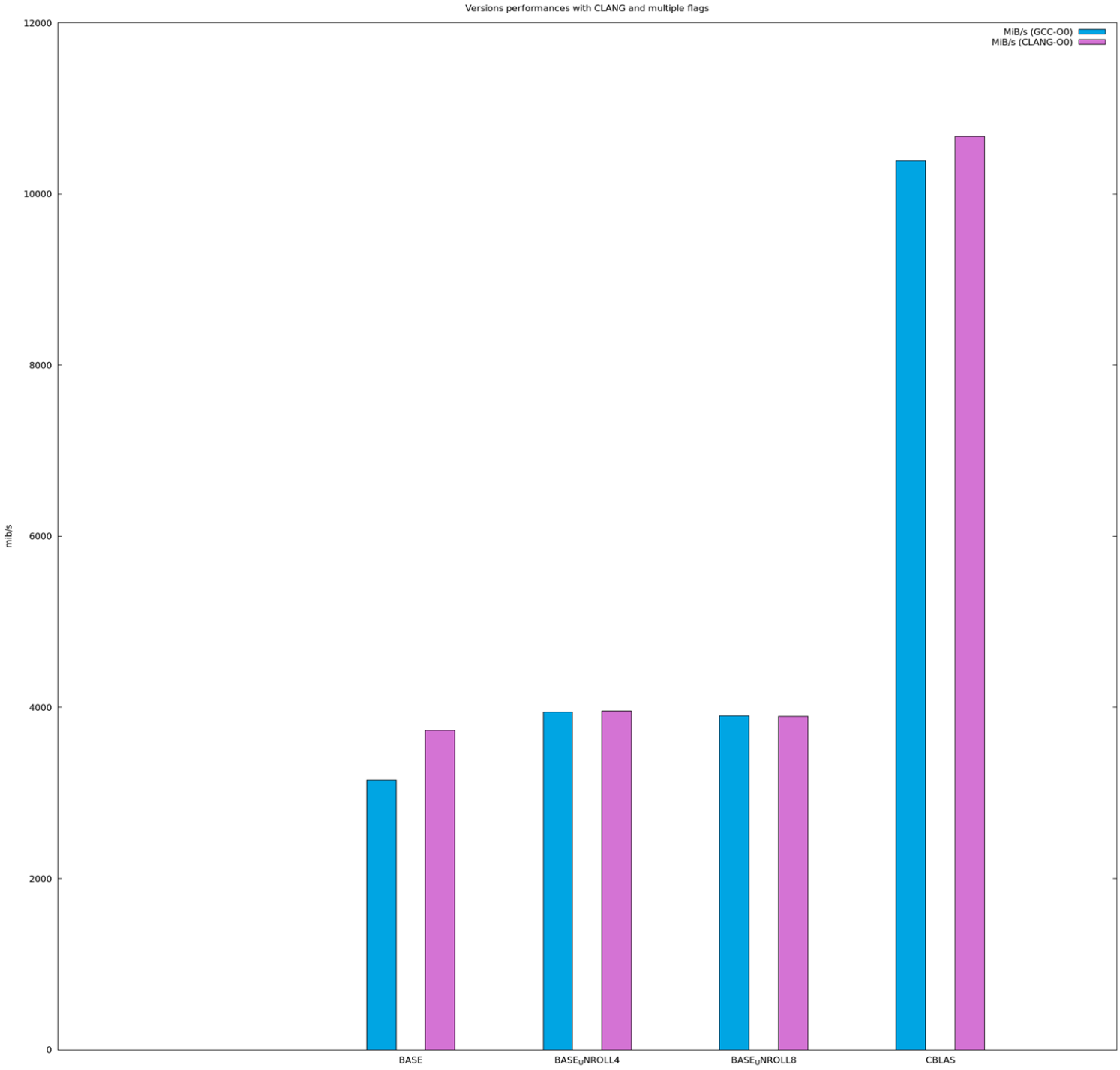
Comparaison entre les versions de CLANG et plusieurs flags :



Pour le compilateur CLANG j'ai obtenus les meilleurs performances avec la version BASE avec le flag -Ofast, toutes les version ont les meilleurs performances avec le flag -Ofast. La dernière performance est obtenus avec BASE avec le flag -O0, qui montre une performance faible aussi pour toutes les versions.

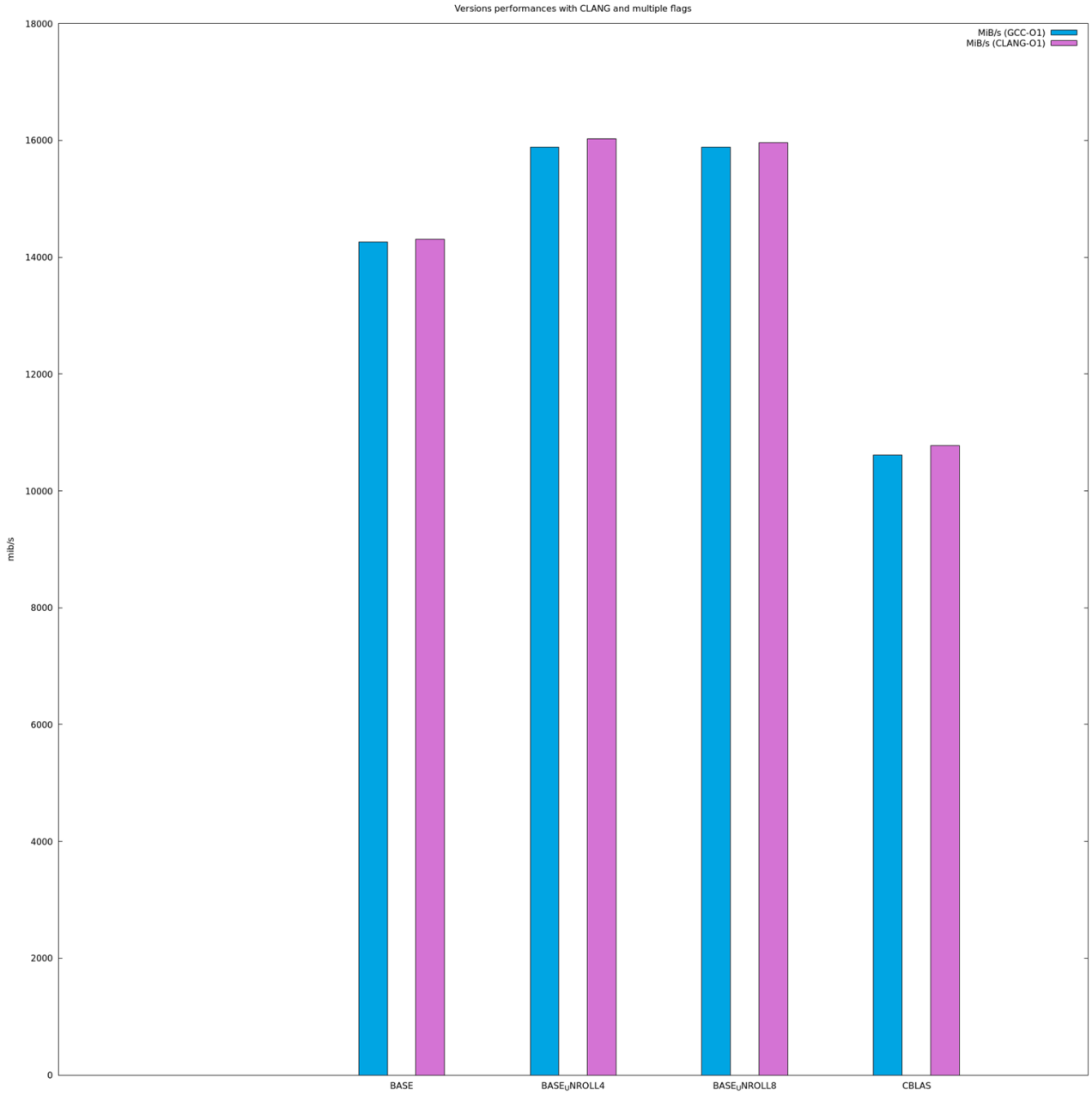
Comparaison entre les version des deux compilateurs :
Comparaison selon chaque flag :

Flag-O0 :



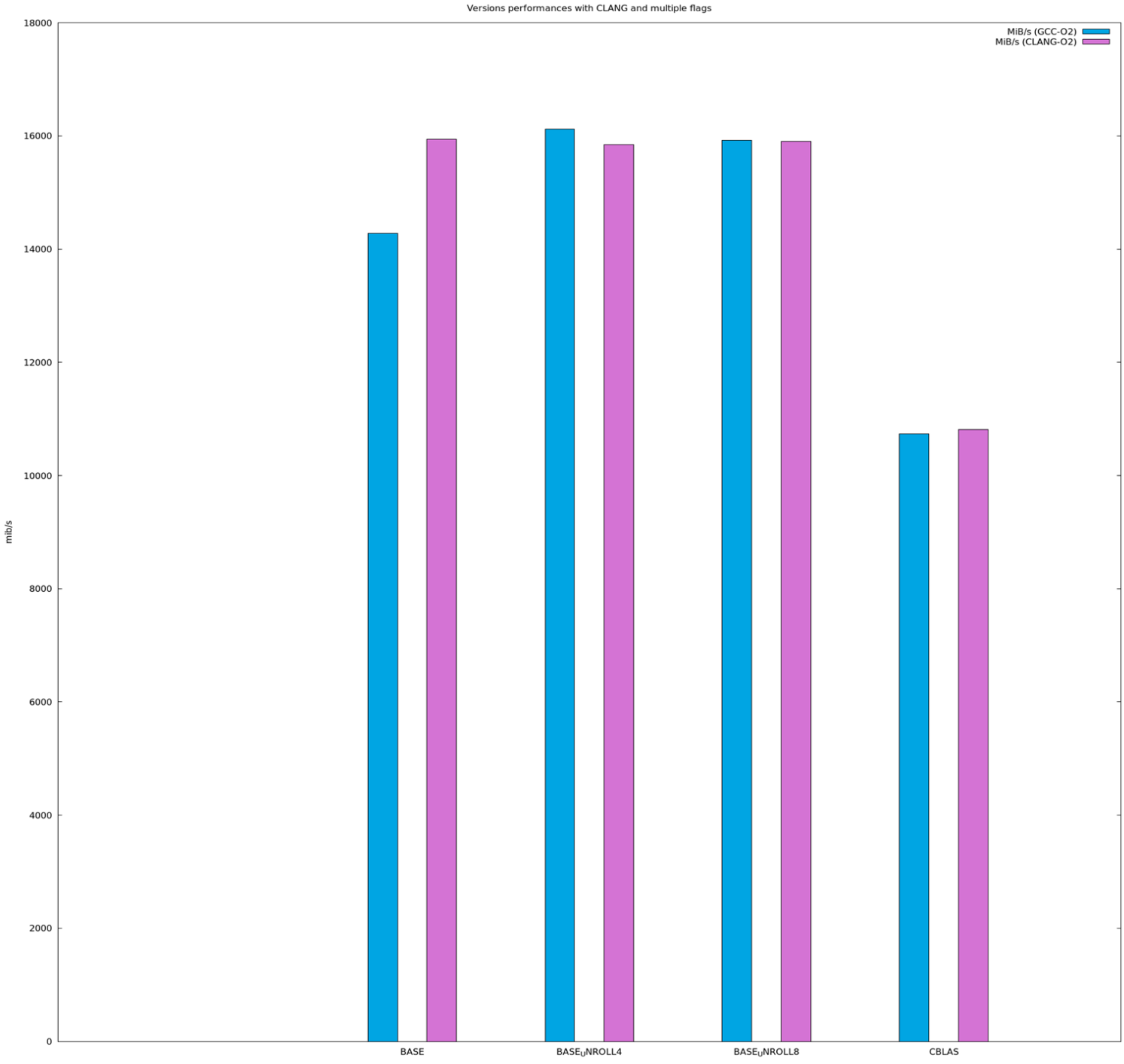
Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version CBLAS et la dernière est obtenus par GCC avec la version BASE. GCC dépasse CLANG avec UNROLL4 et UNROLL8, et CLANG dépasse GCC avec BASE et CBLAS.

Flag-O1 :



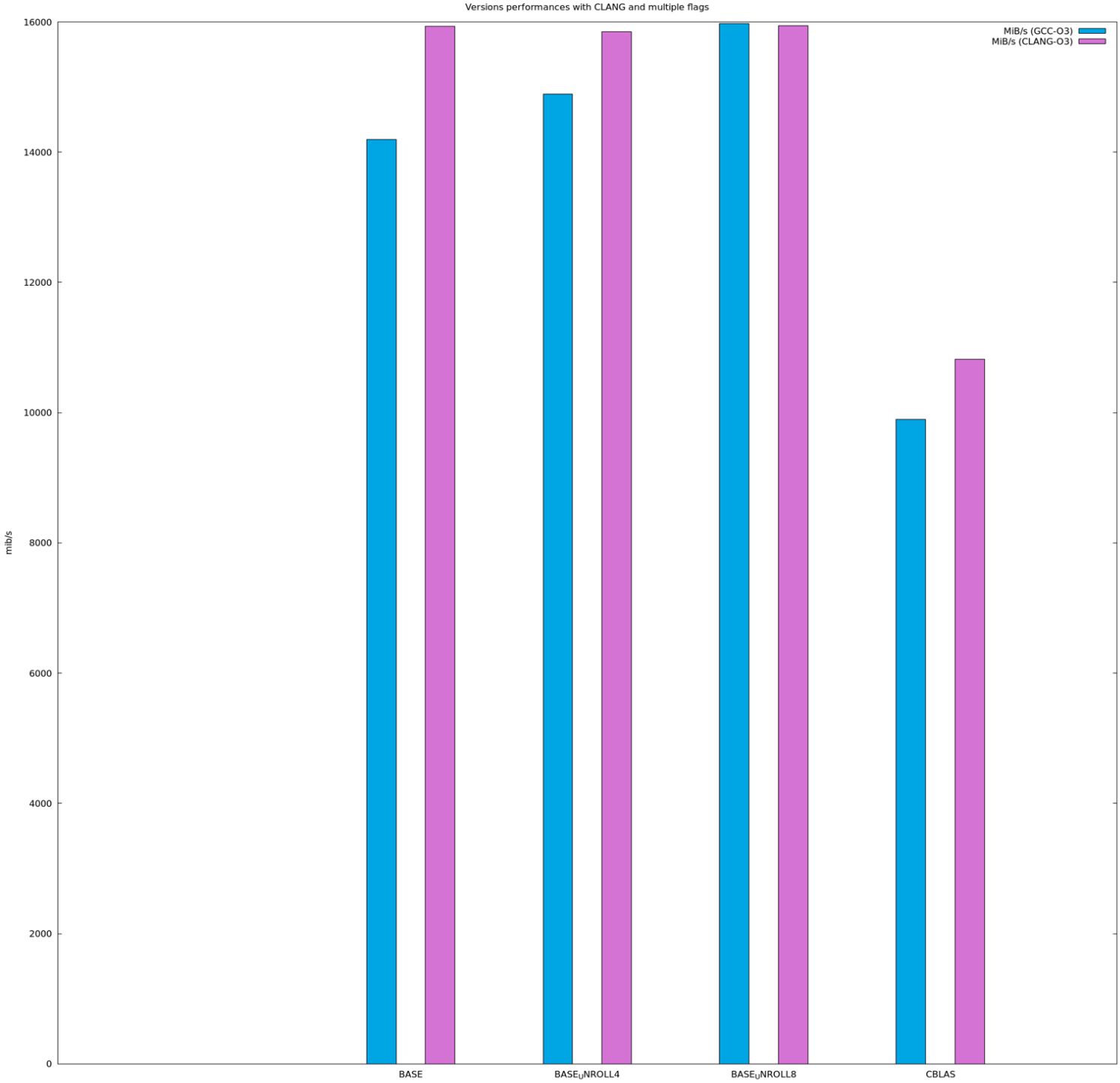
Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version UNROLL4 et la dernière est obtenus par GCC avec la version CBLAS. Toutes les versions de CLANG dépassent les versions de GCC.

Flag-O2 :



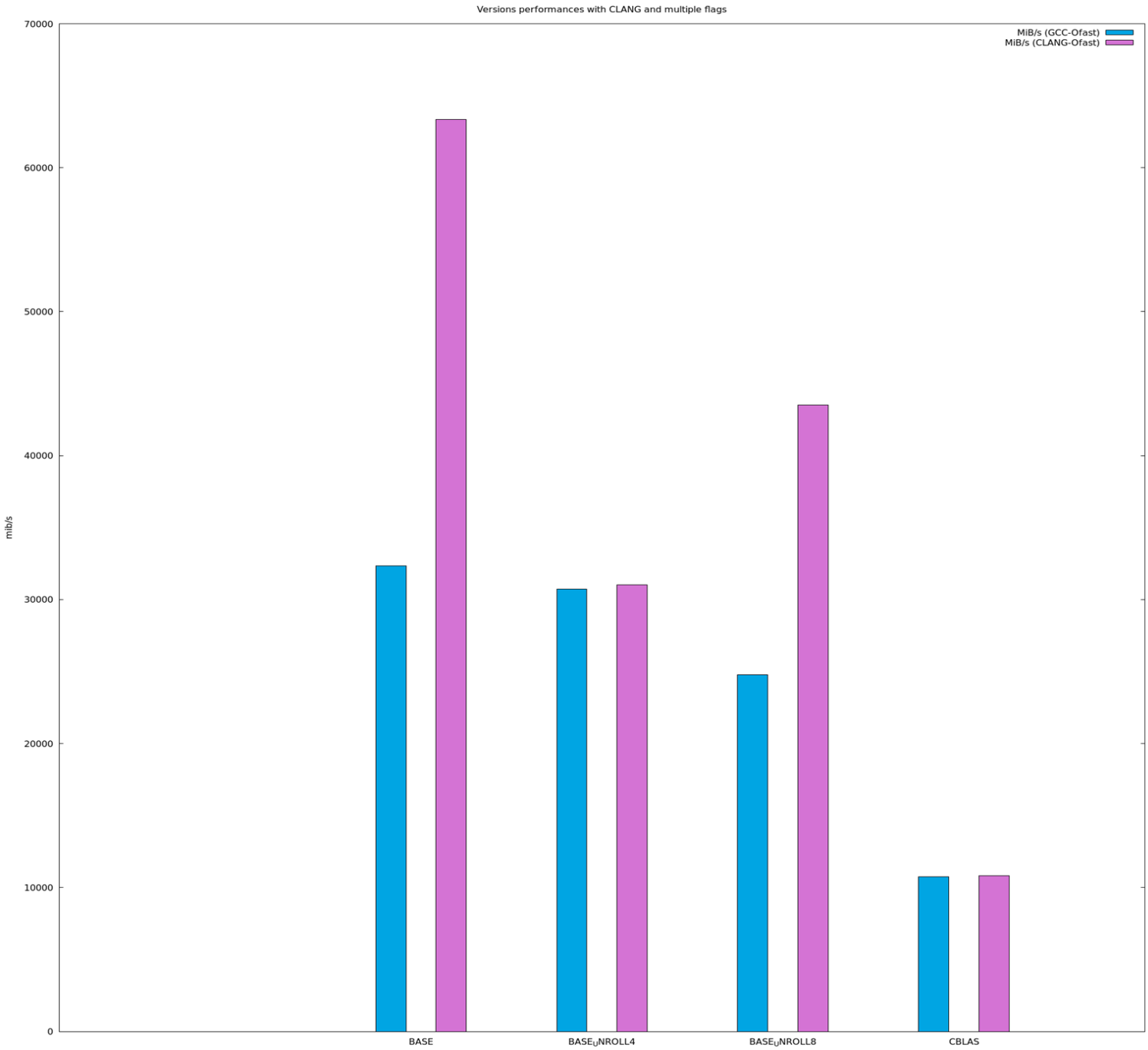
Pour ce flag les meilleures performances sont obtenus par le compilateur GCC avec la version UNROLL4 et la dernière est obtenus par GCC avec la version CBLAS.

Flag-O3 :



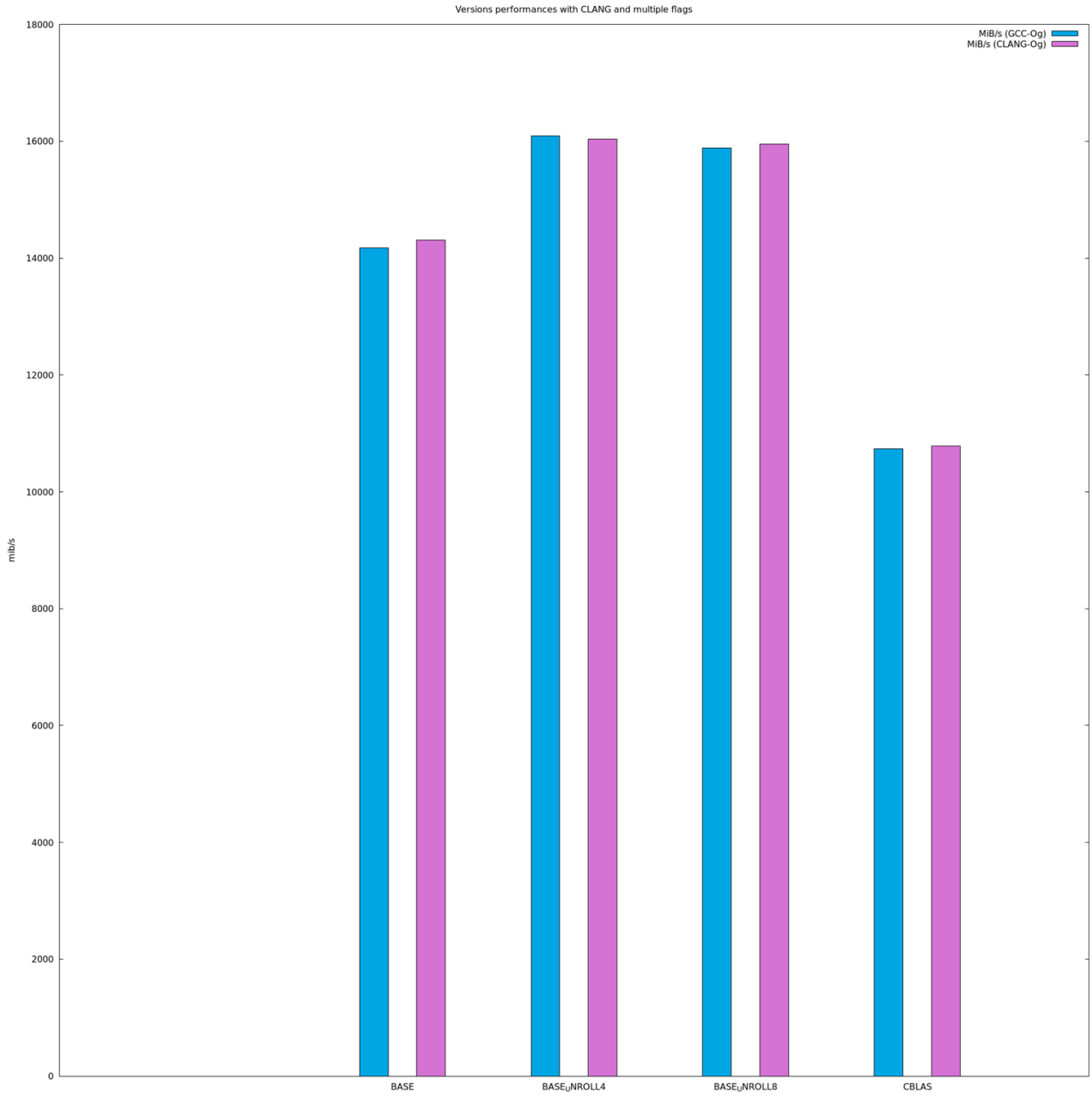
Pour ce flag les meilleures performances sont obtenus par le compilateur GCC avec la version UNROLL8 et la dernière est obtenus par GCC avec la version CBLAS. CLANG dépasse toutes les version de GCC sauf UNROLL8 (GCC-UNROLL8 est mieux que CLANG-UNROLL8).

Flag-Ofast :



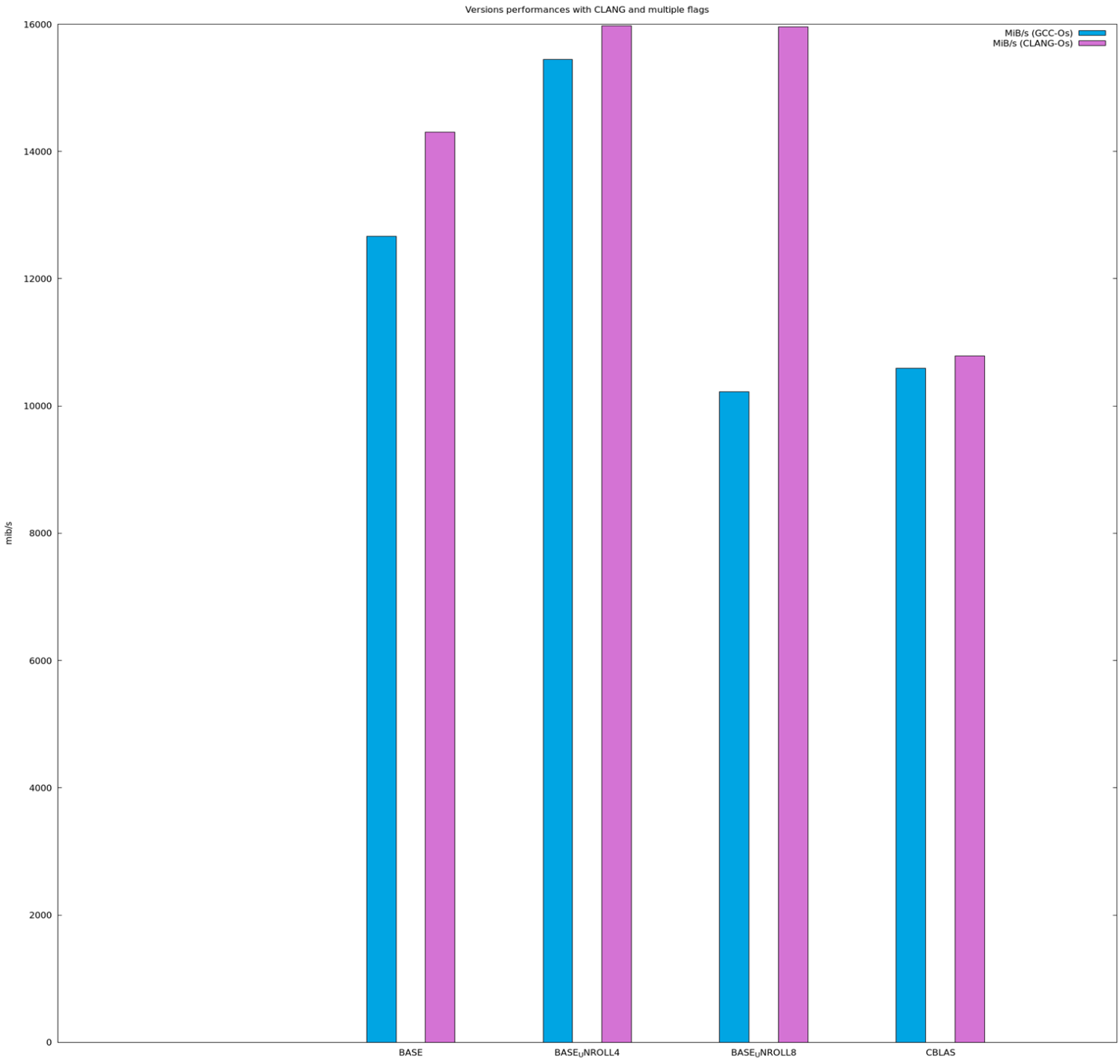
Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version BASE et la dernière est obtenus par GCC avec la version CBLAS. Toutes les versions de CLANG dépassent les versions de GCC.

Flag-Og :



Pour ce flag les meilleures performances sont obtenus par le compilateur GCC avec la version UNROLL4 et la dernière est obtenus par GCC avec la version CBLAS. Toutes les versions de CLANG dépassent les versions de GCC sauf la version UNROLL4 (GCC-UNROLL4 est mieux CLANG-UNROLL4).

Flag-Os :



Pour ce flag les meilleures performances sont obtenus par le compilateur CLANG avec la version UNROLL4 et la dernière est obtenus par GCC avec la version UNROLL8. Toutes les versions de CLANG dépassent les versions de GCC.

Conclusion reduc :

Une analyse similaire à dotprod a été effectuée, avec des versions unroll8, unroll4 et CBLAS. Les performances ont été évaluées de manière similaire avec différentes configurations d'optimisation et deux compilateurs. Les résultats ont également révélé des différences notables entre les versions et les configurations.

CONCLUSION GÉNÉRALE:

L'impact du choix du compilateur et des options d'optimisation sur les performances des fonctions étudiées est significatif. Les versions avec déroulage de boucle, en particulier unroll4, ont montré de bonnes performances dans de nombreuses configurations. Les performances optimales varient en fonction des fonctions, des compilateurs et des flags d'optimisation. Cette étude souligne l'importance de choisir judicieusement les configurations d'optimisation en fonction du contexte d'utilisation pour obtenir les meilleures performances possibles.

Bibliographie :

Wikipédia
https://linux.die.net/man/
Les cours d’université (université de versailles saint quentin en yvelines)