Tom BUDON

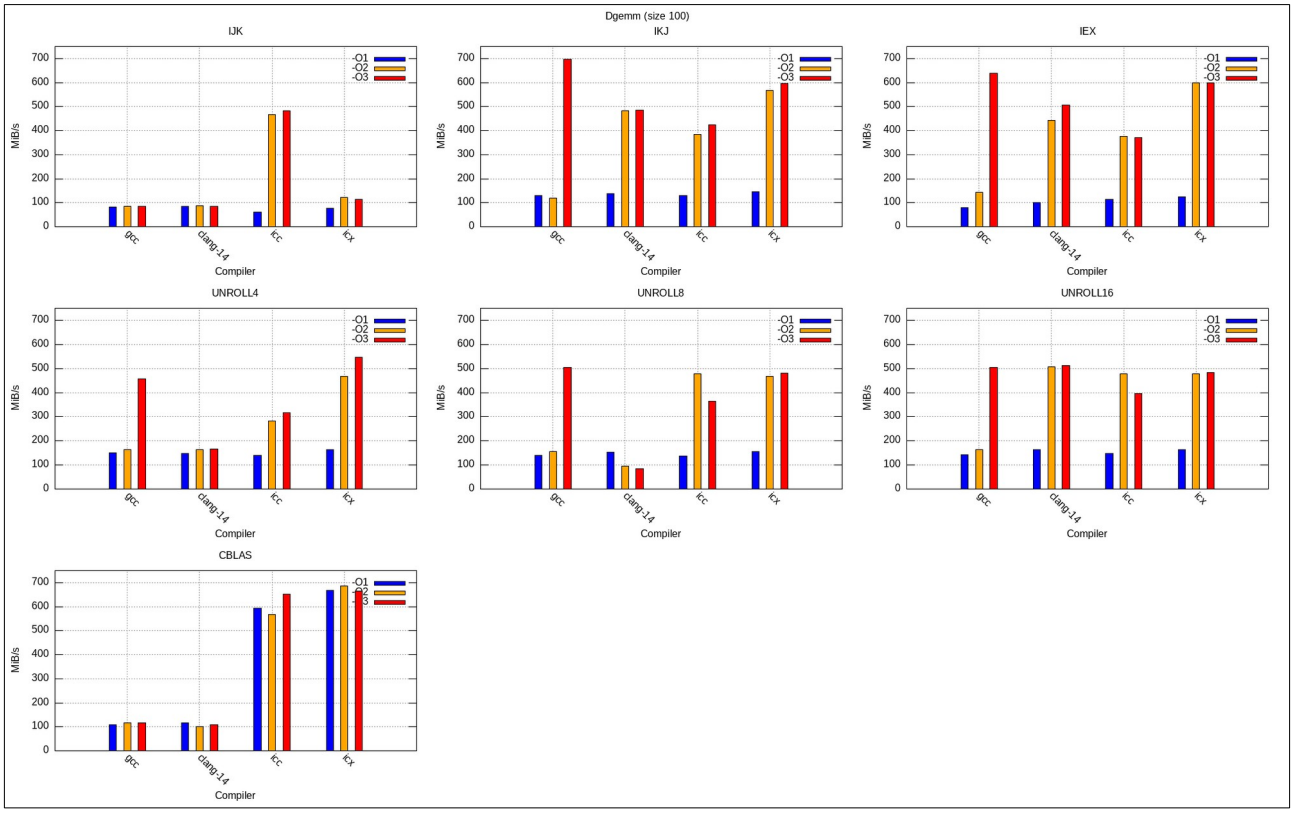
OBHPC RAPPORT

Pour les dgemm j'ai crée un script (Run_benchmark.sh) qui lance le programme dgemm plusieurs fois avec des compilateurs et des flags d'optimisation différents et qui stocke les mesures effectuées dans des fichiers .dat. Ces fichiers sont utilisés par un script gnuplot qui génère plusieurs graphiques.

Le script lance le programme dgemm pour 4 compilateurs (gcc, clang, icc et icx) et 3 flags de compilations (-O1, -O2, -O3), puis récupère les mesures de performances pour toutes les versions de la dgemm (IJK, IKJ, IEX, UNROLL4, UNROLL8, UNROLL16 et CBLAS).

Pour simplifier les benchmarks, j'ai fais en sorte que le code boucle sur une liste avec les noms des compilateurs dedans, à chaque compilateur le script compile le code et l'exécute (3 fois pour O1, O2 et O3) ensuite pour chaque implémentations (IJK, IKJ, …) les résultats sont stockés dans des fichiers .dat grâce à un grep + cut. Par la suite, il sera plus simple d'ajouter un compilateur ou une implémentation grâce à ce code (et plus simple de le copier-coller pour dotprod et reduc).

Les 1ers graphiques générés avec gnuplot représentent les résultats de chaque implémentations de la dgemm pour chaque compilateurs et flags de compilation.



Graphiques dgemm par version

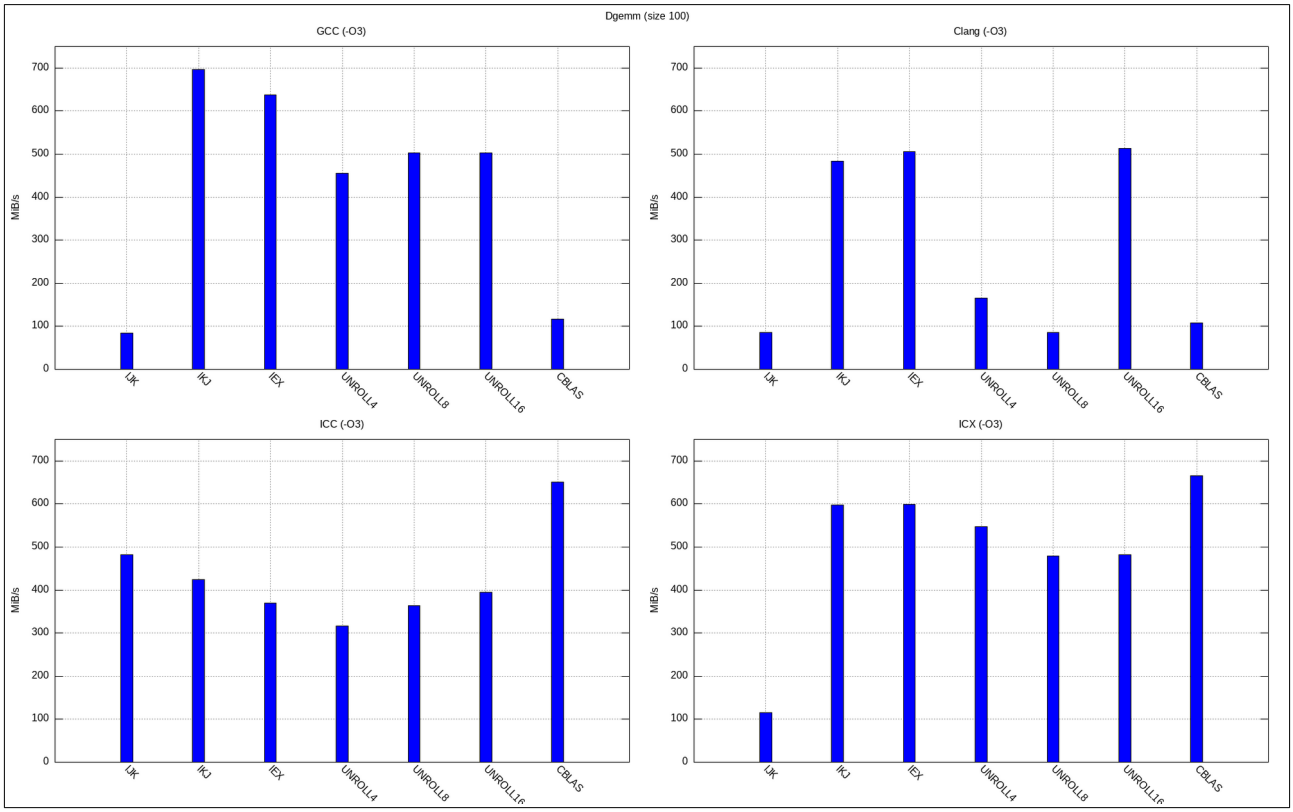
Pour la version IJK les compilateurs gcc, clang et icx n'arrive pas vraiment à optimiser le code, même en O3 par contre icc arrive avec O2 et O3 à avoir de meilleurs résultats. Il détecte probablement que l'on veut faire une dgemm et la remplace par du code à lui déjà optimisé.

Pour les versions IKJ et IEX les compilateurs arrivent mieux à les optimiser avec les flags d'optimisation. Le code de IKJ et IEX est probablement plus simple à unroll pour les compilateurs que celui de IJK.

Pour les trois versions d'unroll gcc et icx sont assez constant, j'imagine qu'ils transforment l'unroll 4 et 8 en 16. Par contre clang a des mauvaises performances sur l'unroll 4 et 8 mais revient au niveau de gcc et icx sur l'unroll 16, donc gcc et icx ont bien dû modifier leurs unroll 4 et leurs unroll 8 en unroll 16. Et de son côté icc a de meilleur performance avec O2 que O3, donc j'imagine qu'il a encore remplacé le code ou juste qu'il fait n'importe quoi.

La version CBLAS est à peine meilleur que celle de base pour gcc et clang, alors que icc et icx obtiennent de bonnes performances avec, même si l'écart-type monte aux alentours de 300%.

Les graphiques suivant représentent les résultats de chaque compilateurs (avec O3) pour chaque implémentations de la dgemm.

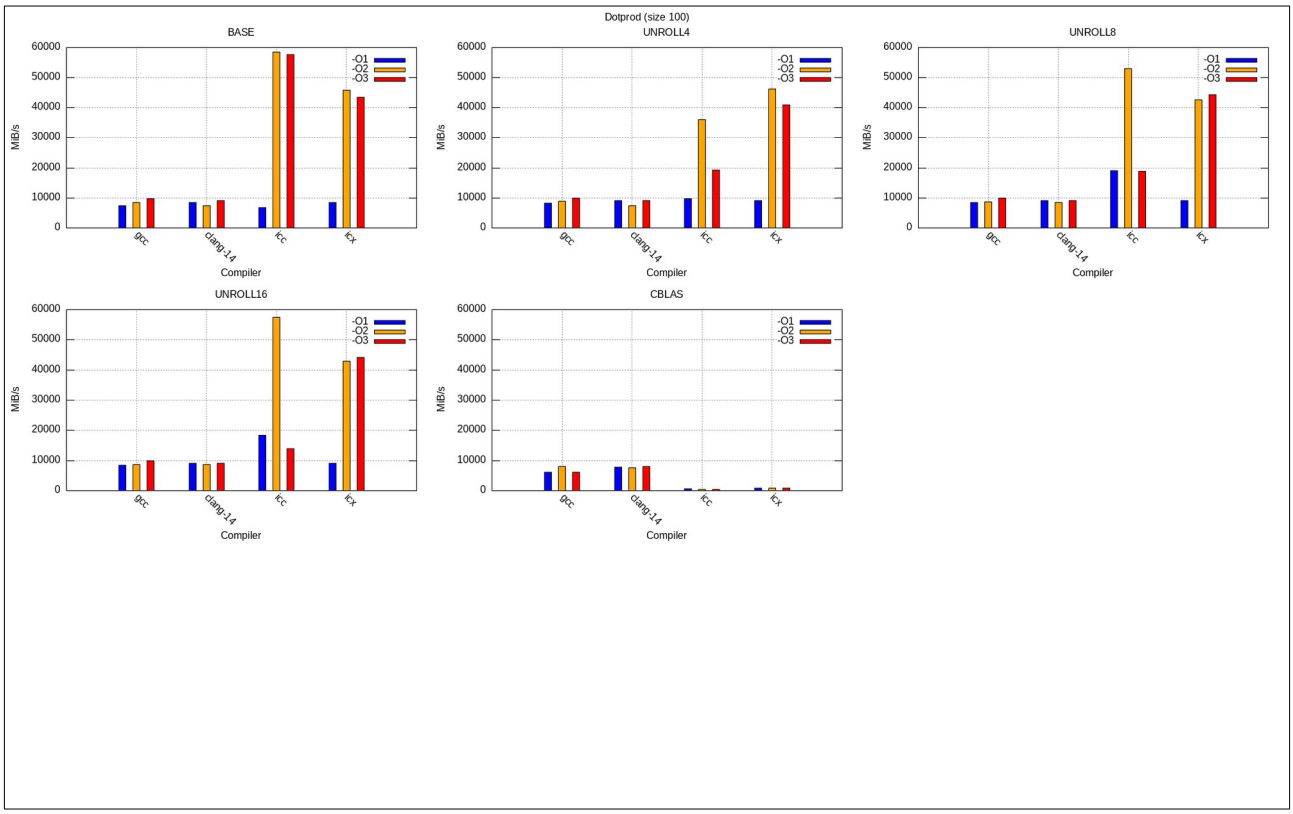


Graphiques dgemm par compilateur

On peux mieux comparer les différentes implémentations pour un même compilateur, et on peut voir que en O3 les versions d'unroll ne sont pas meilleur que IKJ et IEX, sûrement parce que les compilateurs font de l'unrolling sur IJK et IEX alors qu'ils n'y arrivent pas avec IJK.

Dans l'ensemble les mesures ont l'air plutôt stable (à part CBLAS avec icc/icx), pour gcc et icx l'écart-type est globalement < 5%, pour clang et icc certaines mesures montent jusqu'à 15% - 30%.

Pour le dotprod j'ai repris les mêmes scripts bash et gnuplot que ceux pour dgemm.

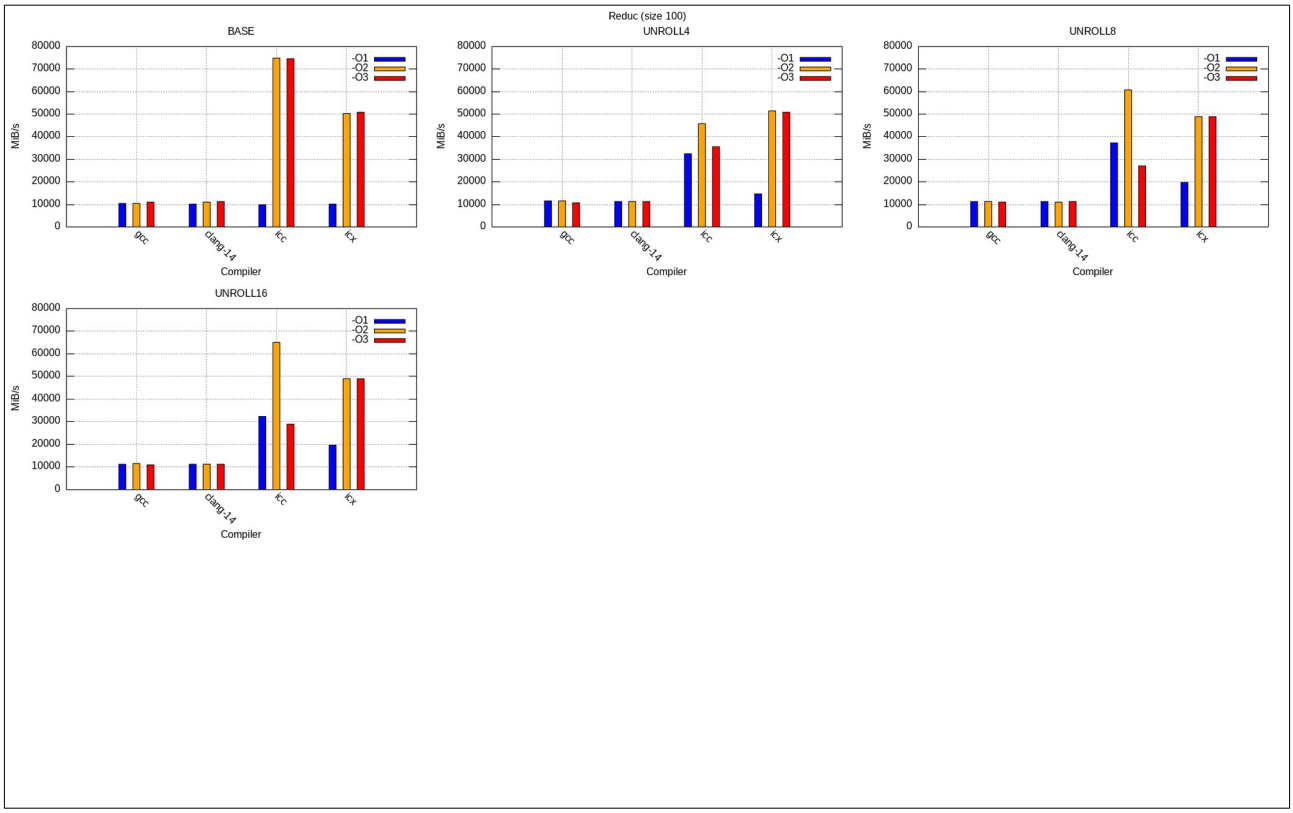


Graphiques dotprod par version

Pour la version de BASE et les 3 versions d'unroll gcc et clang sont assez constant, j'imagine que la version de base, l'unroll 4 et l'unroll 8 transforme en unroll 16 vu qu'elles ont toutes les mêmes performances. Par contre du côté d'icc et icx, on a un X6 en vitesse. Icc doit remplacer le code pour l'implémentation de base mais pas pour les unroll. Et icx est assez stable peut importe la version.

Pour CBLAS, gcc et clang sont un peu moins bon que pour la version de base. Et pour icc et icx les résultats me paraisse bizarre (j'ai du mal utiliser la fonction ddot de blas).

Pour la reduc j’ai repris les mêmes scripts bash et gnuplot que ceux pour dgemm et dotprod.



Graphiques reduc par version

Même constat pour la reduc que pour le dotprod, icc et icx font du X5/X7 par rapport à gcc et clang. Et gcc, clang et icx sont plutôt constant peu importe la version, alors que icc fait n’importe quoi de son côté.