
Projet AOA

*Measuring/Evaluating the performance of a
computer system using benchmarks*

BRINGUIER Maxence

1 février 2021

Table des matières

I- Introduction.....	3
II- Exécution	4
III- Analyse des Benchmarks	5
Copy.....	5
Dotprod	6
Load	7
Memcpy.....	8
Ntstore.....	9
Pc	10
Reduc.....	11
Store	12
Triad.....	13
IV- Conclusion	14

Table des illustrations

Figure 1 : Istopo	3
Figure 2 : Résultat du benchmark copy avec les caches L1, L2, L3.....	5
Figure 3 : Résultat du benchmark dotprod avec les caches L1, L2, L3	6
Figure 4 : Résultat du benchmark load avec les caches L1, L2, L3	7
Figure 5 : Résultat du benchmark ntstore avec les caches L1, L2, L3.....	9
Figure 6 : Résultat du benchmark reduc avec les caches L1, L2, L3	11
Figure 7 : Résultat du benchmark store avec les caches L1, L2, L3	12
Figure 8 : Résultat du benchmark triad avec les caches L1, L2, L3.....	13

I- Introduction

Le but de ce projet est d'exécuter différents benchmarks afin de les comparer et d'en évaluer les performances par la suite. Il est donc préférable d'avoir un minimum de processus s'exécutant sur la machine pour avoir des valeurs propres.

Vous retrouvez ci-dessous les différentes informations concernant les spécifications techniques et matérielles de la machine sur laquelle les benchmarks furent lancées.

Les informations systèmes peuvent être retrouvées dans le dossier `./système/`

Type	Informations
Système d'exploitation	Pop!OS 20.10 x86 64
Processeur	Intel(R) Core(TM) i5-7300HQ
Nombres de cœurs	4 cœurs physiques
Nombres de thread	8
Fréquence	2,5 GHz

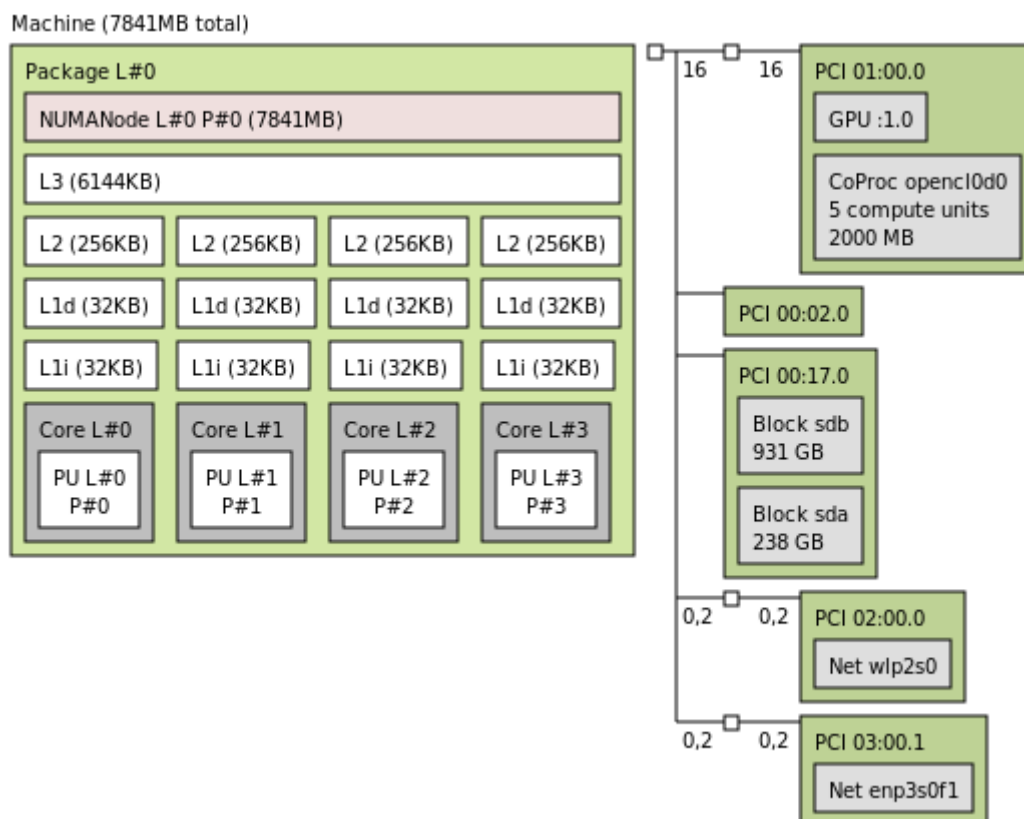


Figure 1 : Istopo

II- Exécution

Afin d'exécuter ces codes il suffit de lancer la commande `./build.sh`.

Le script `build.sh` lancera les exécutions suivantes :

- Compilation de chaque *makefile*
- Exécution des benchmarks en fonction des différents caches L1, L2, L3
- Génération des graphiques en fonction des résultats obtenus

Les valeurs utilisées pour les caches sont les suivantes :

- L1 : 20 KiB
- L2 : 200 KiB
- L3 : 1000 KiB

Chaque benchmark est exécuté sur le cœur 3.

Les valeurs des cœurs et des caches peuvent être modifiées dans le script pour correspondre à votre machine.

III- Analyse des Benchmarks

Copy

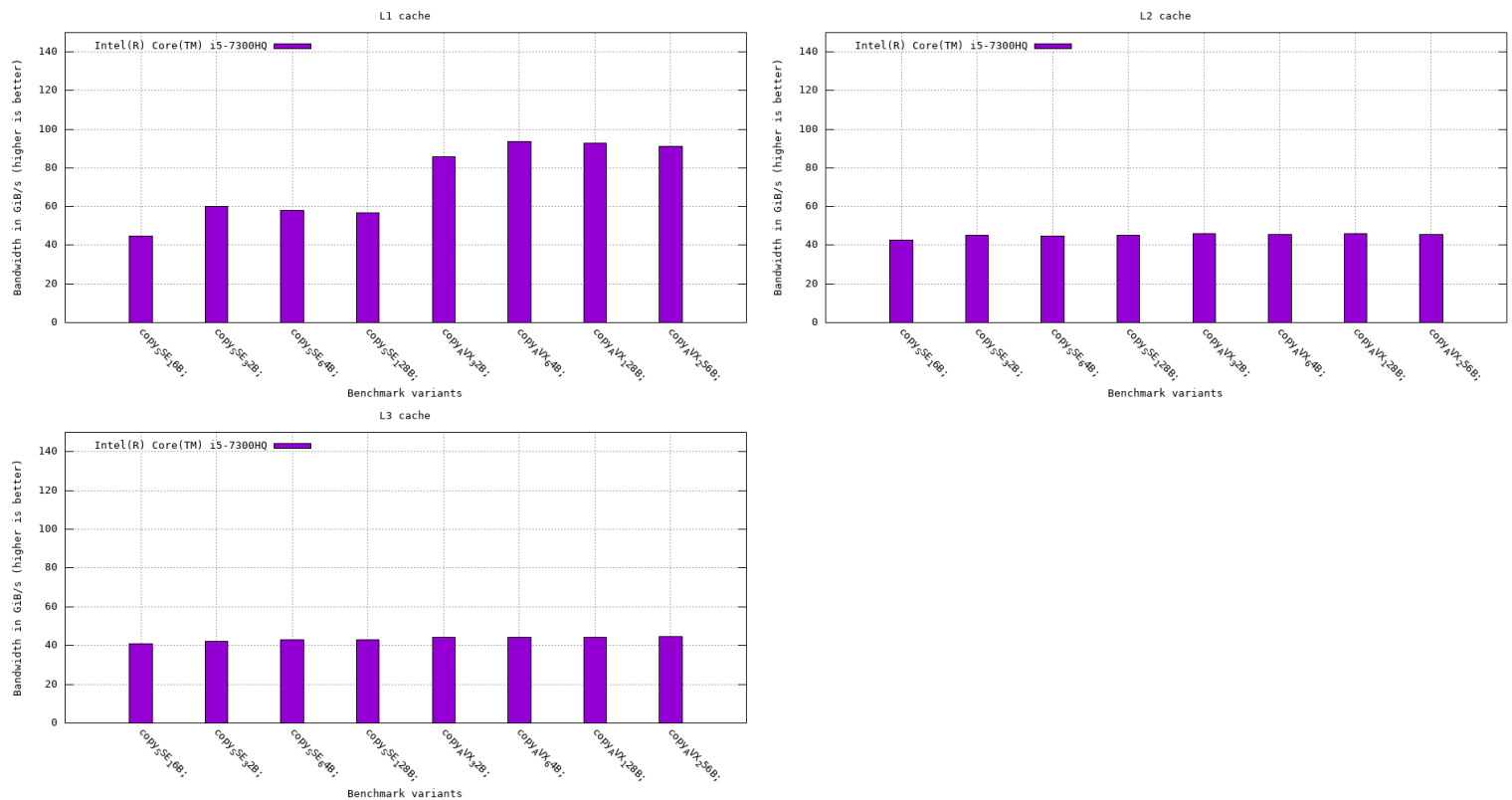


Figure 2 : Résultat du benchmark copy avec les caches L1, L2, L3

Pour le cache L1 on remarque une différence notable entre les instructions SSE et AVX, la bande passante étant de 90 GiB/s pour AVX contre 45 GiB/s pour SSE.

Nous pouvons remarquer une bande passante de 40 GiB/s pour les caches L2 et L3 que ce soit avec des instructions SSE ou AVX.

Dotprod

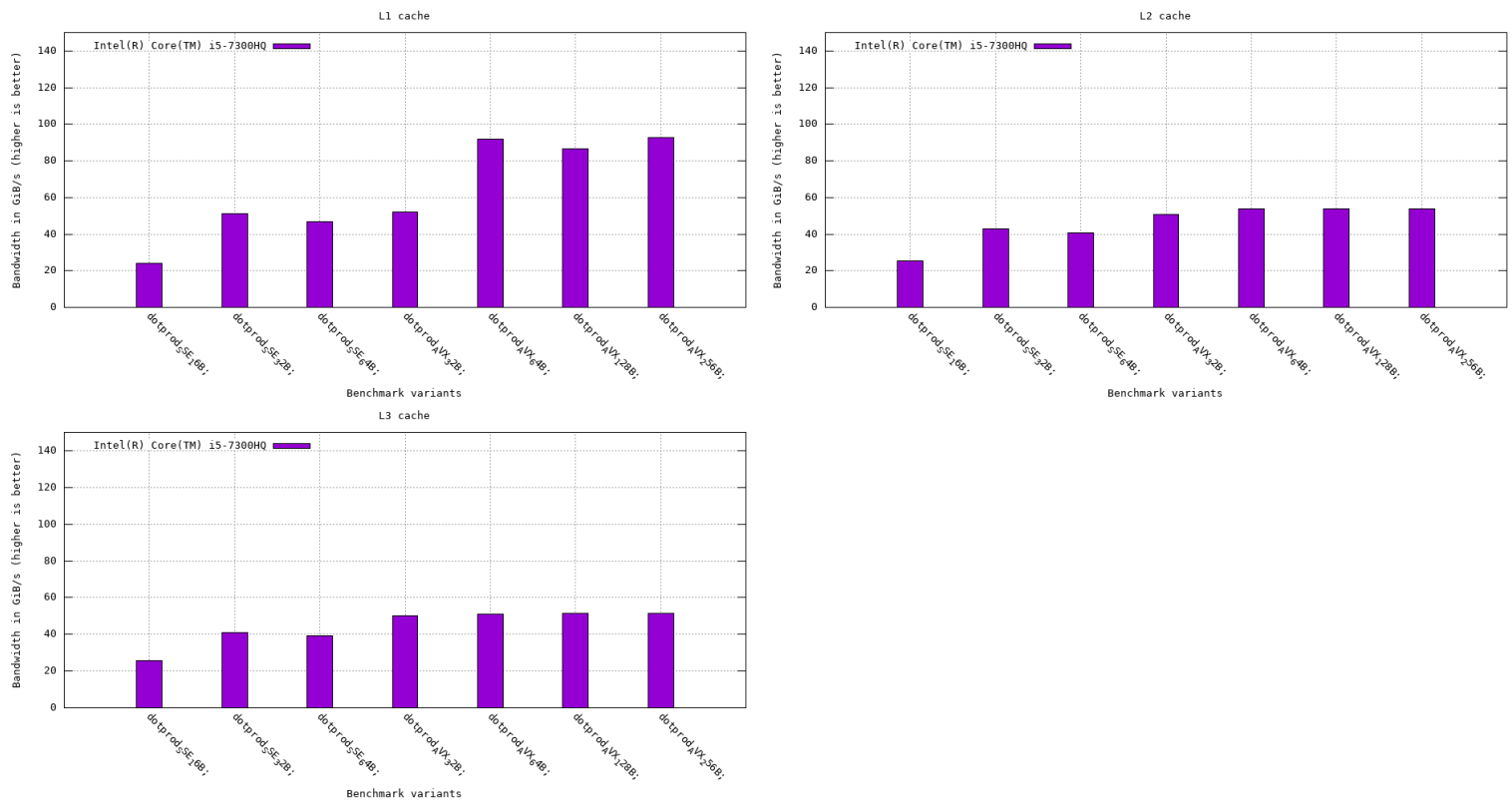


Figure 3 : Résultat du benchmark dotprod avec les caches L1, L2, L3

De la même façon que le *benchmark copy*, les résultats du cache L1 utilisant les instructions AVX sont bien meilleures que celles utilisant SSE. La bande passante est de 90 GiB/s pour AVX contre 45 GiB/s pour SSE.

Nous pouvons remarquer une bande passante de 40 GiB/s pour les caches L2 et L3 que ce soit avec des instructions SSE ou AVX.

Load

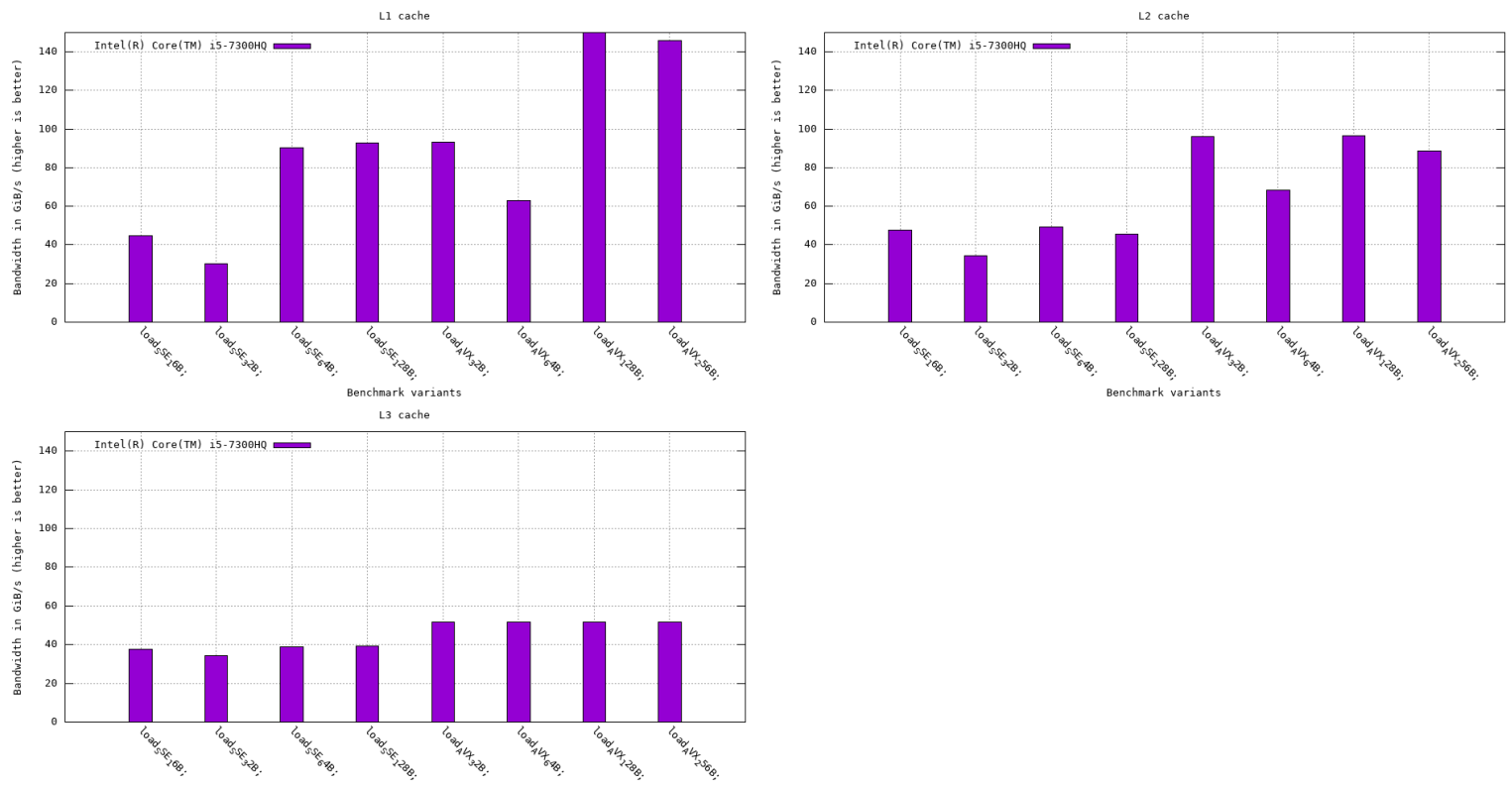
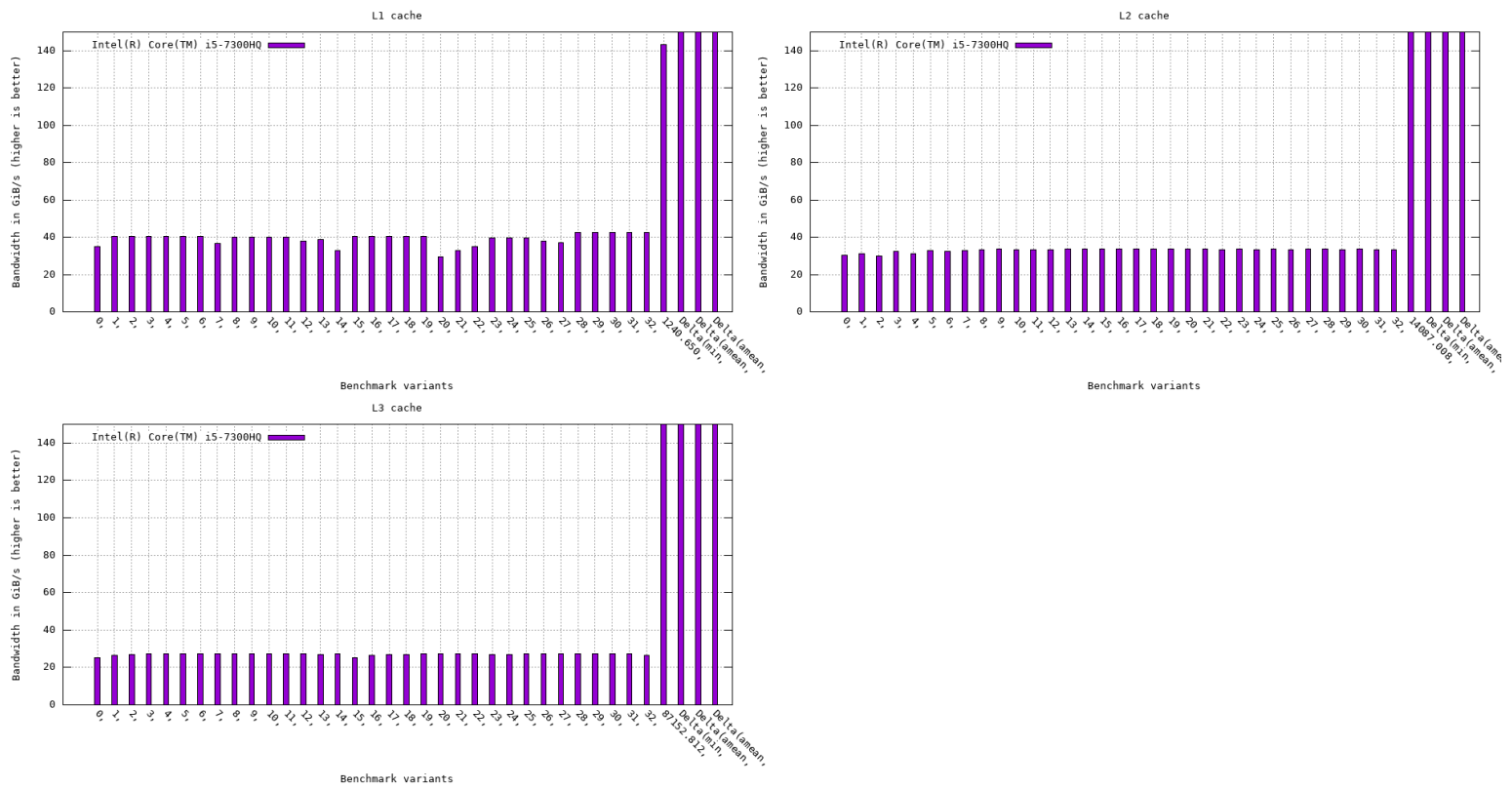


Figure 4 : Résultat du benchmark load avec les caches L1, L2, L3

Pour cette opération les valeurs utilisant les instructions AVX paraissent bien meilleures que les instructions SSE pour les caches L1, L2. La bande passante avec les instructions AVX double presque par rapport aux instructions SSE.

Pour le cache L3, aucunes différences notables ne peuvent être faites.

Memcpy



Les opérations sur le cache L1 sont meilleures que sur les autres caches ; L2 étant également meilleure que L3.

Ntstore

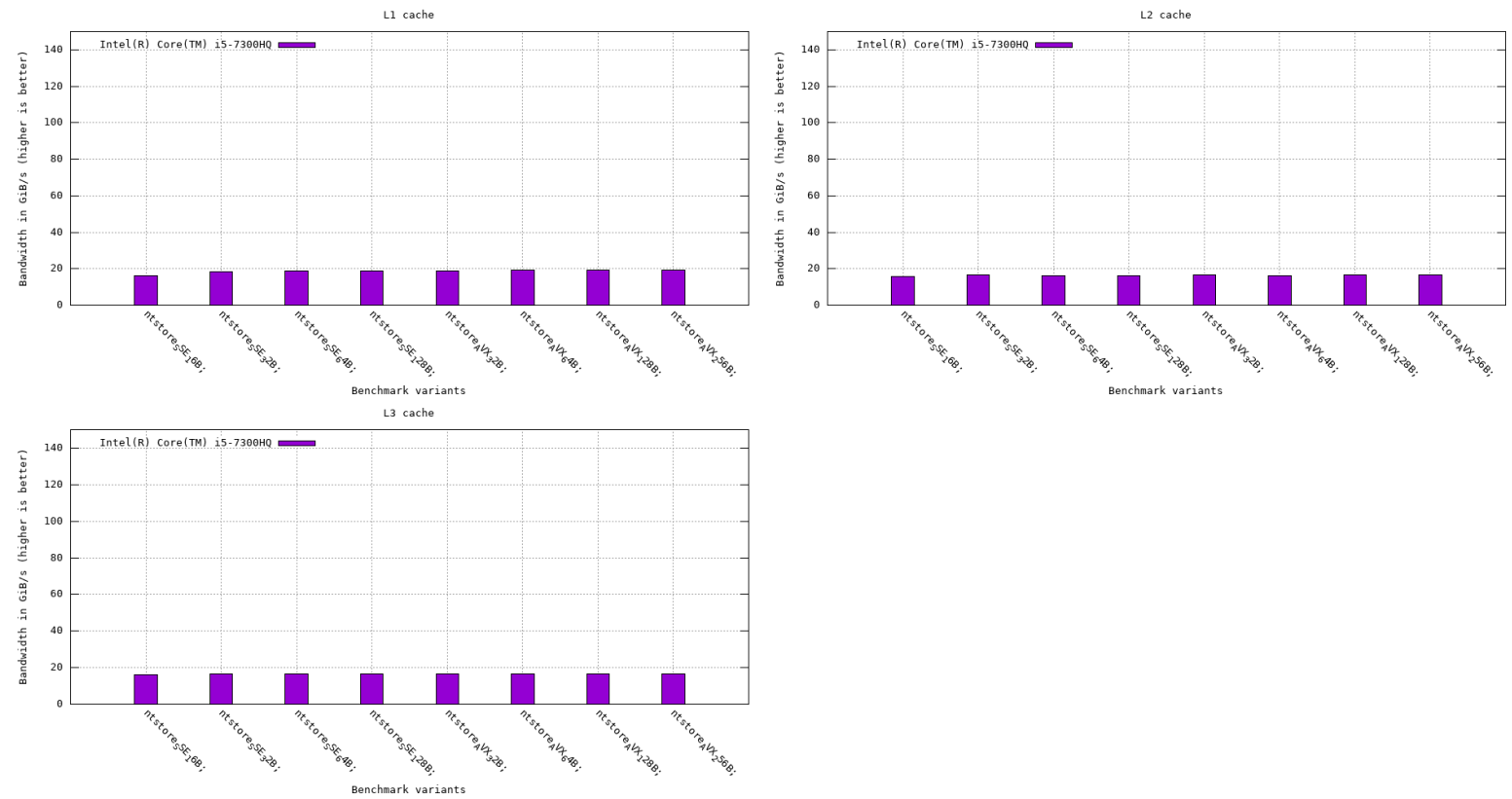
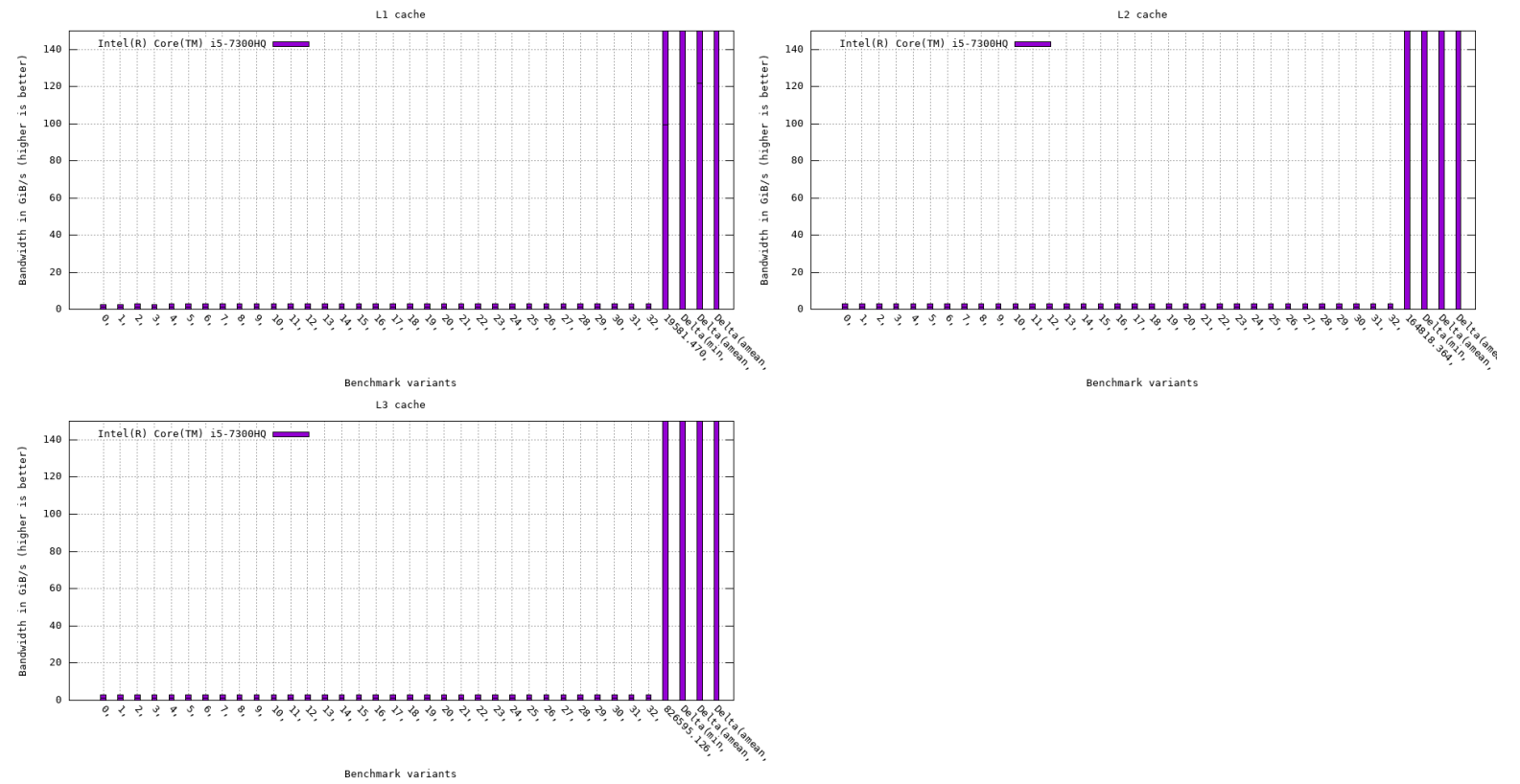


Figure 5 : Résultat du benchmark ntstore avec les caches L1, L2, L3

Cette opération n'utilise pas les caches mais écrit directement dans la DRAM. De ce fait les performances des caches sont similaires.

Pc



Ce benchmark evalue la latence des loads de la DRAM. Il est donc logique de trouver le même résultat pour chacun des caches puisqu'aucun d'eux n'est utilisé.

Reduc

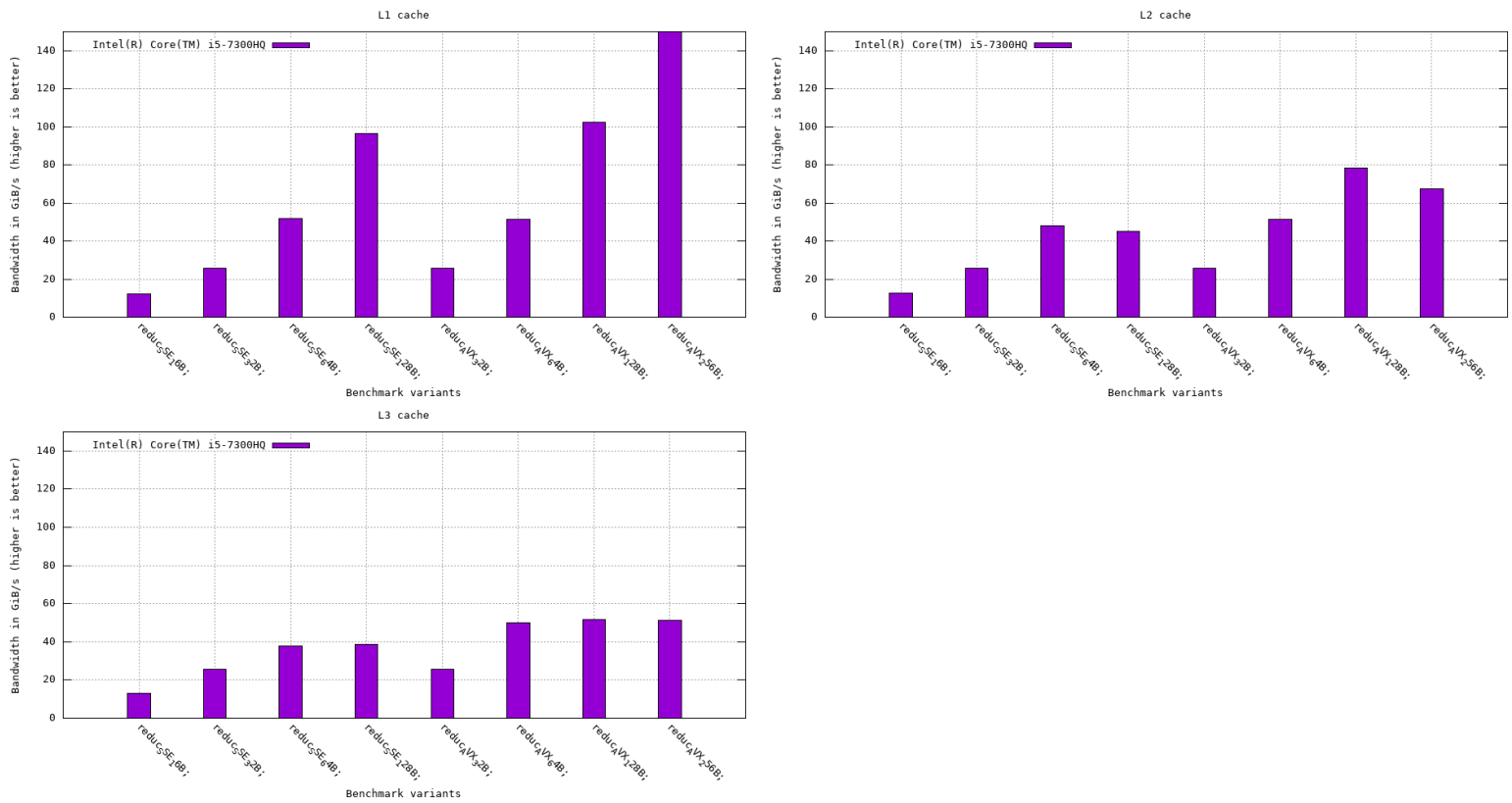


Figure 6 : Résultat du benchmark *reduc* avec les caches L1, L2, L3

Une fois de plus les instructions AVX sont plus performantes que les instructions SSE respectivement pour chaque cache.

Store

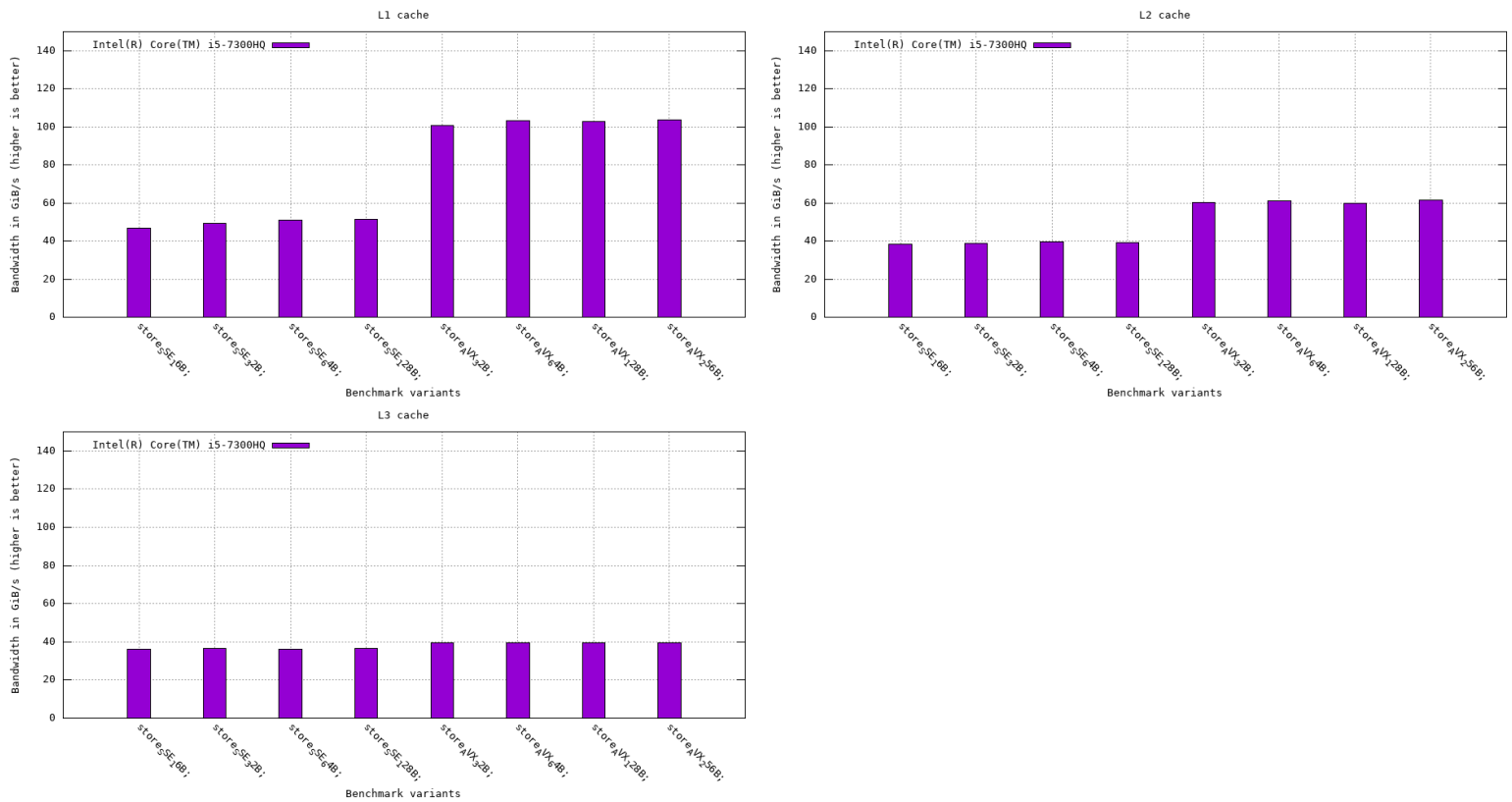


Figure 7 : Résultat du benchmark store avec les caches L1, L2, L3

Les opérations sur le cache L1 sont meilleures que sur les autres caches ; L2 étant également meilleure que L3.

Les instructions AVX sont plus performantes que les instructions SSE.

Triad

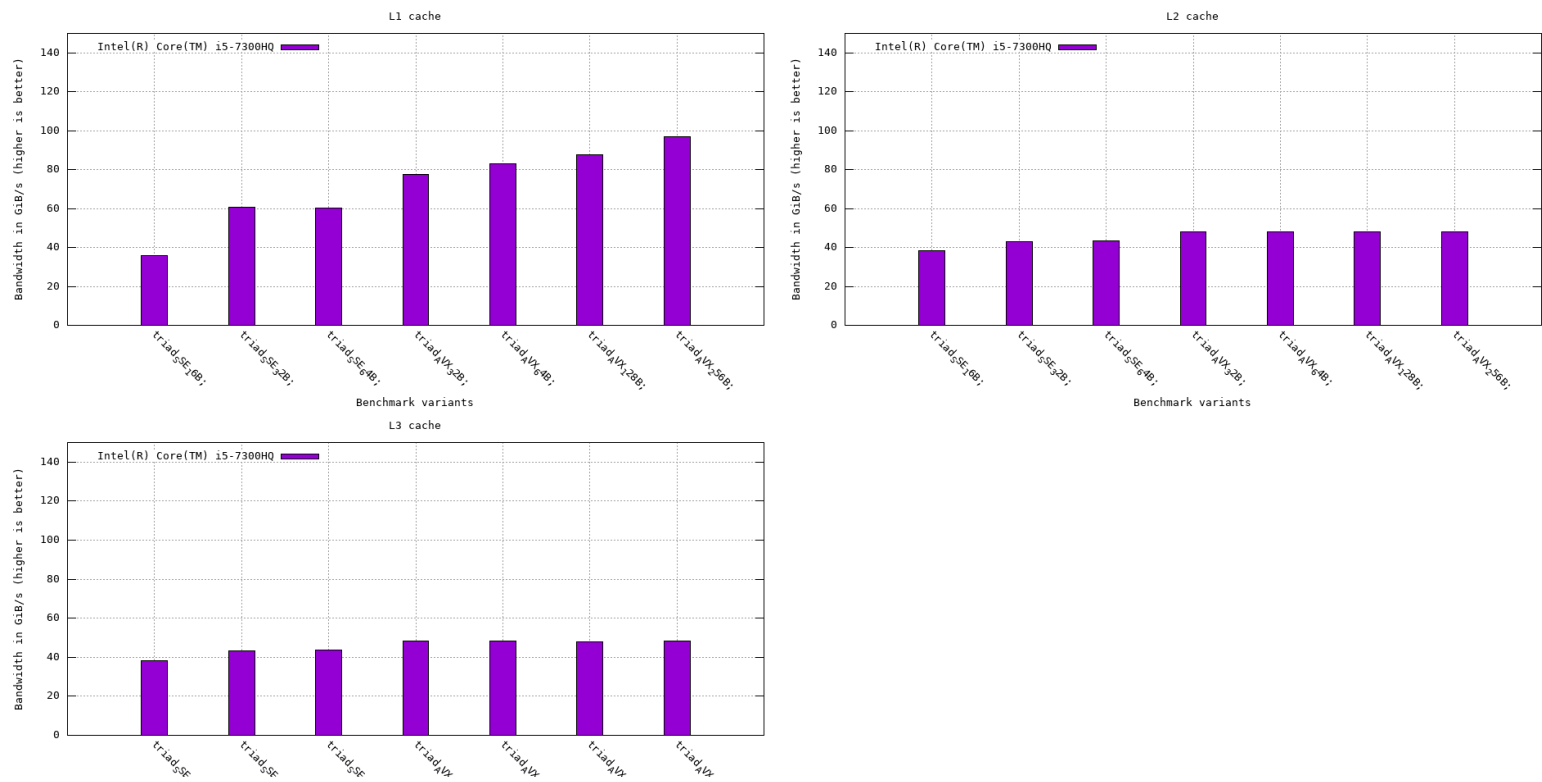


Figure 8 : Résultat du benchmark triad avec les caches L1, L2, L3

Les opérations sur le cache L1 sont meilleures que sur les autres caches. L'écart entre les instructions AVX et SSE est moins flagrant que sur les autres opérations.

IV- Conclusion

Ce projet consistait à exécuter des benchmarks sur la mémoire de la machine. Nous avons donc pu observer les taux d'erreur et les performances selon le cache utilisé.

Comme attendu, nous avons vu que les performances du cache L1 sont plus rapides que le cache L2, lui-même plus rapide que le cache L3.

Nous avons pu également nous rendre compte que les instructions AVX ont de meilleurs résultats/bande passante que les instructions SSE.