

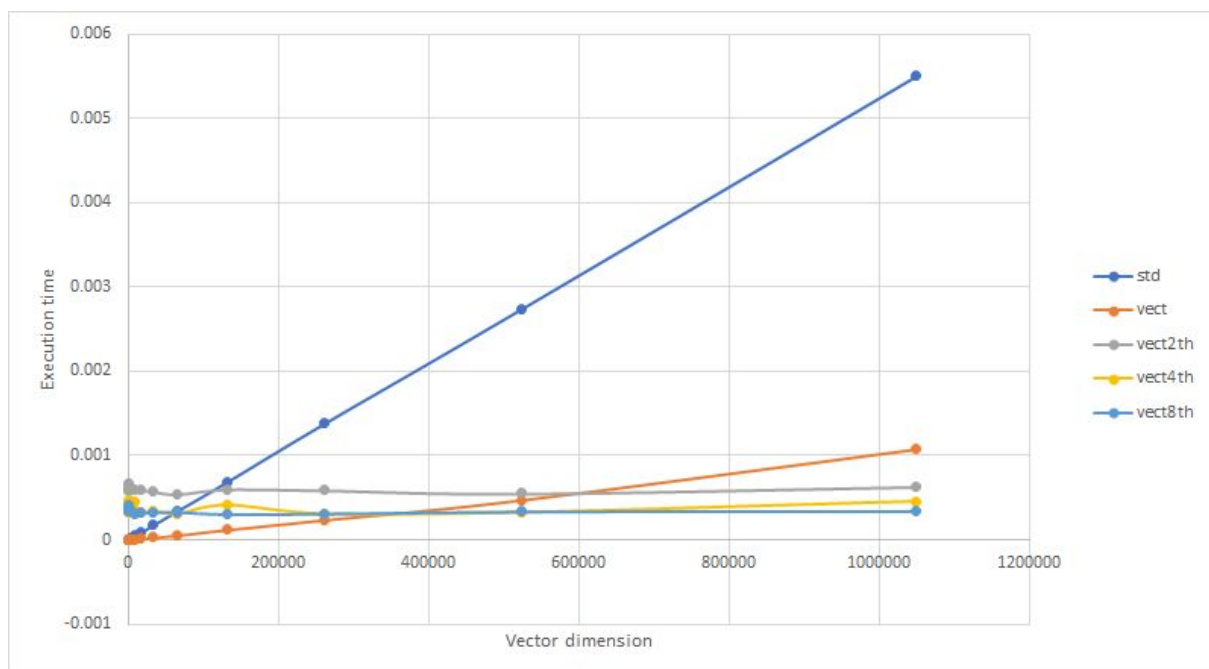
## Rendu du Projet - Architecture Matérielle et Logicielle des Ordinateurs

*Analyse de la performance de l'algorithme de calcul de la norme par rapport aux différentes méthodes de calcul utilisées*

Daniel Al Choboq

Simon Saliba

*Variation du temps d'exécution en fonction de la dimension du vecteur d'entrée*



Le graph ci-dessus est obtenu en utilisant  $n = 2^{15}$  (1048576).

Pour des dimensions faibles, les méthodes non multithreadées sont plus efficaces en temps de calcul que les méthodes multithreadées puisque ces dernières ont besoin d'un temps initial pour initialiser et créer les threads. Dans ce cas, le temps de calcul est inférieur au temps de création des threads seul.

La méthode vectorielle est 5,1 fois plus rapide que la méthode standard.

Théoriquement, celle-ci doit être 8 fois plus rapide. Ceci est dû au fait que dans l'implémentation de la méthode vectorielle, nous utilisons une commande en plus que dans la méthode standard :

- Dans la méthode standard, la valeur absolue est calculée en utilisant la fonction `abs(n)`
- Dans la méthode vectorielle, la valeur absolue est calculée en utilisant les 2 fonctions `_mm256_sqrt_ps(_mm256_mul_ps(v,v))`

La méthode multi threadée (à 2 threads) devient plus efficace que la méthode standard à partir d'une dimension de 130 000, et plus efficace que la méthode vectorielle à partir de 600 000.

La méthode à 4 threads est 1,7 fois plus rapide que la méthode à 2 threads, ce qui est logique. Théoriquement, elle doit être 2 fois plus rapide.

Les méthodes à 4 et 8 threads sont presque confondues. Ceci est dû au fait que la machine sur laquelle nous réalisons ce test ne possède que 4 cœurs.

**Donc la méthode vectorielle à 4 threads est 15x plus rapide que la méthode standard. Théoriquement, celle-ci doit être  $8 \times 4 = 32$  fois plus rapide.**