



***Compte rendue***

présenté par

**Idriss HASSINE**

***Informatique Multi coeur***

Encadrant

Mr. Domas STEPHANE

# Mesures et Interprétations

## 1.1 Interprétation des mesures

On suppose que seul un processus (le processus 0) a accès au contenu donc on s'intéresse aux mesures de temps fournies par le processus 0. Le fichier myhost utilise seulement 3 machines du cluster : cluster1, cluster3 et cluster4.

	2*2	3*3	4*4	5*5
100	0.027856	0.038319	0.089743	0.169450
200	0.166438	0.222424	0.559172	0.944466
300	0.496195	0.702004	1.710638	2.943112
400	1.173551	1.707362	4.107505	6.996012
500	2.219179	3.205208	9.326524	14.337718

TABLE 1.1 – Canon

	2*2	3*3	4*4	5*5
100	0.029203	0.034399	0.078782	0.164479
200	0.158048	0.214237	0.550062	0.927919
300	0.492121	0.691974	1.670046	2.916157
400	1.164187	1.689269	4.083260	7.509658
500	2.216019	3.198797	9.412617	14.701099

TABLE 1.2 – Snyder

	2*2	3*3	4*4	5*5
100	0.0241448	0.032096	0.077851	0.137481
200	0.145282	0.208864	0.514959	2.814708
300	0.459220	0.659373	1.690524	2.814708
400	1.120025	1.624635	4.155581	6.449969
500	2.143842	3.140545	9.309549	14.232285

TABLE 1.3 – Fox

	2*2	3*3	4*4	5*5
100	0.025193	0.042650	0.078297	0.189097
200	0.149608	0.241383	0.524988	1.100039
300	0.473777	0.756290	1.559342	2.969721
400	1.147731	1.806148	4.044405	6.977977
500	2.176988	3.377013	9.170161	15.320750

TABLE 1.4 – Pblas

L'algorithme pblas donne des résultats importants (dans le sens de mettre un temps plus réduit) quand on manipule des blocs de grande taille.

Pour des blocs de petite taille L'algorithme de Fox donne des mesures plus performantes pour des blocs de petite taille.

Pour des blocs de grande taille : pour des grilles de petite taille l'algorithme de Fox est le plus meilleur mais lorsque la taille de grille augmente ce résultat n'est pas encore vrai car l'algorithme de Pblas commence à donner des résultats plus meilleurs.

Un résultat me semble anormal est celle de bloc de taille 500 et la grille de 5\*5 pour l'algorithme de Pblas : 15.320750 s

Cette valeur doit être inférieure aux autres valeurs fournies par les autres algorithmes. Car l'algorithme de Pblas n'effectue pas la rotation de la matrice B à chaque multiplication locale.

## 1.2 le temps d'exécution divisé par le nombre de processeurs

Les mesures suivantes correspondent au processus 0.

Le fichier myhost utilise seulement 3 machines du cluster : cluster1, cluster3 et cluster4.

Taille de grille	2*2	Taille de grille	3*3	Taille de grille	4*4
150	0.02015675	100	0.00449077777	75	0.0028374375
300	0.1240765	200	0.0244	150	0.0158351875
450	0.400865	300	0.07718033333	225	0.049970375
600	0.9740825	400	0.18913844444	300	0.10614925

TABLE 1.5 – Canon

Taille de grille	2*2	Taille de grille	3*3	Taille de grille	4*4
150	0.01921025	100	0.00401355555	75	0.0026713125
300	0.119535	200	0.02386344444	150	0.0143813125
450	0.3934595	300	0.07666844444	225	0.046313
600	0.96464275	400	0.18800866666	300	0.106737

TABLE 1.6 – Snyder

Taille de grille	2*2	Taille de grille	3*3	Taille de grille	4*4
150	0.0163905	100	0.003539	75	0.00219875
300	0.11394575	200	0.02323122222	150	0.0134235
450	0.38362475	300	0.07361955555	225	0.04586325
600	0.94991875	400	0.18181655555	300	0.104660875

TABLE 1.7 – Fox

Taille de grille	2*2	Taille de grille	3*3	Taille de grille	4*4
150	0.017967	100	0.00471222222	75	0.00239475
300	0.118571	200	0.02713311111	150	0.0145338125
450	0.39102975	300	0.08389111111	225	0.0472389375
600	0.96327725	400	0.20071755555	300	0.0960454375

TABLE 1.8 – Pblas

Pour des blocs de petites taille toujours l’algorithme de fox est plus rapide et donne des valeurs plus petites et ceci est bien expliqué par le fait que l’algorithme de fox n’a pas de problème en termes de communication, par contre cet algorithme est coûteux de point de vue de mémoire. Ce problème n’est plus visible ou observable pour les blocs et les grilles de petite taille donc l’algorithme de fox donne des résultats plus performants dans ce cas.

Pour des blocs de grande taille et des grilles de petite taille l’algorithme de fox continue à être le plus rapide et de donner des valeurs plus petits. Une fois les blocs et les tailles de grilles deviennent plus grands l’algorithme de Fox n’est plus l’algorithme le plus rapide car il va être plus coûteux en matière de mémoire, cette fois-ci l’algorithme de Pblas donne des résultats plus performants et plus rapides et ceci est logique et bien expliqué par le fait que l’algorithme de Pblas ne fait pas tourner la matrice B après chaque multiplication locale effectuée.

Pour les algorithmes du canon et Snyder, ils sont toujours moins efficaces et coûteux en matière de temps de calcul car les opérations de skew ou de transposition nécessitent un temps de calcul supplémentaire.

### 1.3 Résultats pour une seule machine

Dans cette partie on s’intéresse pour l’étude des résultats fournis par une seule machine, pour cette raison on modifie le fichier myhost au lieu d’utiliser 3 machines de cluster comme pour les résultats précédents, cette fois, on utilise une seule machine celle de cluster1. On obtient les résultats suivants pour le processus 0 :

	2*2	3*3	4*4
600	3.690201		
400		4.392817	
300			3.870431

TABLE 1.9 – Canon

	2*2	3*3	4*4
600	3.667665		
400		3.705663	
300			3.073517

TABLE 1.10 – Snyder

	2*2	3*3	4*4
600	3.670368		
400		3.704030	
300			3.134797

TABLE 1.11 – Fox

	2*2	3*3	4*4
600	3.686646		
400		3.695549	
300			3.206034

TABLE 1.12 – Pblas

Commençons par un bloc de taille 600 et deux gride de taille 2\*2, dans ce cas on remarque que pour la première fois l’algorithme de Snyder est la plus rapide ( avec une petite différence par rapport à l’algorithme de fox de l’ordre de 0.003603 s, cette différence est de l’ordre de 0.02 s par rapport l’algorithme de Pblas et de 0.03 par rapport l’algorithme du canon) ceci est explicable par le faite d’utiliser des ressources limitées car cette fois on a utilisé une seule machine de cluster et l’algorithme de fox est connue par une utilisation couteuse en matière de mémoire.

En faisant diminuer la taille de bloc à 400 et augmenter la taille de gride à 3\*3, l’algorithme de plas est le plus rapide cette fois et ceci est attendu car en augmentant la taille de gride Pblas est plus efficace car il ne fait pas tourner la matrice B suite à chaque multiplication locale.

On finit par diminuer la taille de bloc à 300 et augmenter la taille de 4\*4 on remarque que pour la deuxième fois l’algorithme de Snyder est la plus rapide ceci était inattendu de ma part j’ai attendu d’avoir l’algorithme de Pblas le plus rapide sur tout après avoir augmenté la taille de gride. On peut expliquer ce résultat encore une fois par l’utilisation d’une machine de cluster.