

TP1 Méthodes Numériques

Lucas Matthieu

Jérémy Pucci

Eloi Charra

14/02/2022

1 Calculs de performance

Cette première partie consiste à mesurer la puissance de nos machines. Pour cela nous allons calculer la puissance théorique et réelle de nos processeurs respectifs. La puissance d'un processeur se mesure en FLOPS/s (nombre d'opérations flottantes par seconde).

1.1 Performance théorique

Nous avons premièrement calculé la puissance théorique de chacun de nos processeurs. Ce calcul prend en compte la fréquence d'horloge du processeur ainsi que son nombre de coeurs physiques :

$$freq.proc * nbCoeurs * 16 = xGFLOPS/s$$

Voici un tableau résumant nos calculs :

	Fréquence processeur	Nombre de coeurs	GLFOPS/s
Lucas	$1.6 * 10^9$	4	83.2
Jérémy	$2.6 * 10^9$	2	102.4
Eloi	$2.1 * 10^9$	4	134.4

1.2 Performance mesurée

Après un calcul théorique, nous allons tester ces valeurs en lançant un même algorithme sur 3 langages différents. Ce programme réalise une multiplication de matrice pour une matrice de grande taille. Nous prendrons $N = 2048$ comme nombre de colonnes et de lignes de la matrice. Grâce au temps de calcul du programme ainsi que la taille de la matrice nous pouvons approcher une valeur en FLOPS/s de la puissance de l'ordinateur. Nous comparerons ensuite les valeurs calculées via les programmes avec les valeurs théoriques.

1.2.1 Python

	Temps (s)	Température (°C)	Energie CPU (J)	Energie Mémoire (J)
Lucas	2288	49	9778	1001
Jérémy	2714	44	16672	
Eloi	2971	58	6842	

- $PEloi = 2^{34}/2971 = 0.0058GFLOP/S$ Soit 0,006% de la capacité du processeur
- $PJérémy = 2^{34}/2714 = 0.0063GFLOP/S$ Soit 0,006% de la capacité du processeur
- $PLucas = 2^{34}/2288 = 0.0075GFLOP/S$ Soit 0,009% de la capacité du processeur

1.2.2 Java

	Temps (s)	Température (°C)	Energie CPU (J)	Energie Mémoire (J)
Lucas	121	45	464	65
Jérémy	153	57	470	
Eloi	85	51	225	

- $PEloi = 2^{34}/85 = 0.03GFLOP/S$ Soit 0,2% de la capacité du processeur
- $PJérémy = 2^{34}/153 = 0.11GFLOP/S$ Soit 0,11% de la capacité du processeur
- $PLucas = 2^{34}/121 = 0.14GFLOP/S$ Soit 0,17% de la capacité du processeur

1.2.3 C

	Temps (s)	Température (°C)	Energie CPU (J)	Energie Mémoire (J)
Lucas	284	49	733	341
Jérémy	262	56	700	
Eloi	485	44	645	

- $PEloi = 2^{34}/485 = 0.2GFLOP/S$ Soit 0,02% de la capacité du processeur
- $PJérémy = 2^{34}/262 = 0.06GFLOP/S$ Soit 0,06% de la capacité du processeur
- $PLucas = 2^{34}/284 = 0.06GFLOP/S$ Soit 0,07% de la capacité du processeur

Avec -O3 :

	Temps (s)	Température (°C)	Energie CPU (J)	Energie Mémoire (J)
Eloi	54	43	94	

- $PEloi = 2^{34}/54 = 0.32GFLOP/S$ Soit 0,24% de la capacité du processeur

1.3 Conclusion

???

2 TP Polynômes

Lors de ce TP, nous avons du écrire plusieurs fonctions permettant de manipuler des polynômes pleins dans un premier temps et creux par la suite.

2.1 Polynômes pleins

2.1.1 Egalité polynome

La première fonction que nous avons réalisée a été `egalite_polynome`. Pour vérifier si deux polynômes sont égaux, il faut s'assurer que tous leurs coefficients sont égaux. Nous regardons premièrement si leurs degrés sont différents, si c'est le cas, ils sont forcément non égaux. Ensuite une simple boucle `for` nous permet de parcourir tous les coefficients des deux polynômes et regarder s'ils sont égaux.

2.1.2 Addition polynôme

?

2.1.3 Multiplication polynôme scalaire

Pour multiplier un polynôme par un scalaire nous parcourons simplement tous ses coefficients que nous multiplions par le scalaire donné.

2.1.4 Eval polynôme

Pour calculer la valeur d'un polynôme avec une valeur **x** fournie, nous parcourons tous les coefficients et nous réalisons le calcul suivant pour tous ces coefficients : $val = coeff * x^i$. **i** étant le degré du coefficient (variable de la boucle **for**). Il suffit enfin d'additionner tous ces résultats pour avoir la valeur finale.

2.1.5 Multiplication polynômes

La multiplication est une double distribution. Nous imbriquons donc deux boucles **for** qui parcourent respectivement les coefficients des deux polynômes et les multiplient entre eux.

2.1.6 Puissance polynome

Pour réaliser une puissance de polynômes nous utilisons la fonction précédente `multiplication_polynomes` que nous itérons **n** fois sur elle-même.

2.2 Composition polynôme

?

2.3 Polynômes creux

Pour réaliser un polynôme creux nous avons choisi d'ajouter un deuxième tableau dans la structure. Ce nouveau tableau permet de stocker la valeur des degrés respectif aux coefficients dans le premier tableau.

Tableau des coefficients	3	14	1
Tableau des degrés	256	32	2

- Nous avons un tableau de floats pour les coefficients : `[3.0, 14.0, 1.0]`
- Et un tableau d'entier pour les degrés : `[256, 32, 2]`

Ces tableaux représentent le polynôme $3x^{256} + 14x^{32} + x^2$