

Java Performance - TP4

November 13, 2018

Thomas COUCHOUD
thomas.couchoud@etu.univ-tours.fr
Victor COLEAU
victor.coleau@etu.univ-tours.fr



Chapter 1

Introduction

Chapter 2

Exercice 1

On remarque que beaucoup de types utilisés des classes (Float, Integer, ...), il serait donc peut-être plus efficace d'utiliser des types primitifs directement afin d'éviter d'inutiles opérations d'unboxing/boxing.

De plus nous avons changé les "k = k + 1" en "k++".

Après des mini benchmarks nous obtenons la sortie suivante:

Listing 2.1 – benchmark.txt

	Benchmark	(i)	Mode	Cnt	Score	Error	Units
1	Exercice1.test	0	thrpt	10	70006.601	+= 1883.562	ops/s
2	Exercice1.test	1	thrpt	10	71369.058	+= 1905.356	ops/s
3	Exercice1.testA	0	thrpt	10	457780.134	+= 6152.443	ops/s
4	Exercice1.testA	1	thrpt	10	467063.858	+= 7418.504	ops/s

Avec nos améliorations on peut constater que le score est multiplié par environ 5. Cela est probablement dû aux changements des types plus qu'à la transformation des incréments.

Chapter 3

Exercice 2

La première remarque est que dans le calcul d'une valeur de fibonacci, le calcul des deux précédentes est nécessaire. Or dans l'implémentation donné `fibonacci(i-1)` et `fibonacci(i-2)` sont indépendantes alors que `fibonacci(i-1)` utilise et donc recalcule lui-même `fibonacci(i-2)`. Ce dernier calcul est donc effectué au moins 2 fois, ce qui est inutile et peut être très long sur des grandes valeurs.

Chapter 4

Exercice 3

4.1 Exercice 3a

4.2 Exercice 3b

Chapter 5

Exercice 4

Chapter 6

Exercice 5

Chapter 7

Conclusion