

PAP

# **Projet : rapport3**

4TIN804U

BERASATEGUY Tanguy, GOEDEFROIT Charles

2021-2022

## Table des matières

<b>1</b>	<b>4.5 AVX implementation</b>	<b>2</b>
1.1	4.5.1 The synchronous case . . . . .	2
1.2	4.5.2 The asynchronous case . . . . .	2
<b>2</b>	<b>4.7 OpenCL Implementation</b>	<b>2</b>
2.1	4.7.1 Basic OpenCL Implementation . . . . .	2

## 1 4.5 AVX implementation

### 1.1 4.5.1 The synchronous case

On fait le speedup avec *omp\_tile* entre les tiling *opt* et *avx* sur la machine *UHURA* on obtient 4483 pour *opt* et 4764 pour *avx* un speedup de  $0.94 = \frac{4483}{4764}$ . Il n'y a pas une grande différence entre les 2 version car gcc a une très bonne vectorization.

### 1.2 4.5.2 The asynchronous case

On a implémenter en suivant la consigne sur sujet. La version *avx* fonction avec la variante *omp\_tiled* avec un speedup de TODO : Par-contre la version *avx* ne fonctionne pas avec la variante *omp\_lazy*.

Le code de la fonction :

---

1 `toto`

---

## 2 4.7 OpenCL Implementation

### 2.1 4.7.1 Basic OpenCL Implementation

sur la machine *troi* on a executer la version ocl avec une taille de 1024x1024 et des tuiles de taille 16x16, les 69191 iterations sont executer en 1648ms.

sur la machine *troi* on a executer la version *omp\_tiled* avec une taille de 1024x1024 et des tuiles de taille 16x16 et le tuilage *opt*, cela ce fini au bout de ...ms en ... iterations.

le speedup et de  $4.81 = \frac{332871}{69191}$ .