# AAD - Assignment 1

107474-Joseane Pereira
109050-Gabriel Costa
Universidade de Aveiro, DETI

November 27, 2024

# Contents

# 1 Introduction

In this assignment we were tasked with implementing search functions for a cryptocurrency, **"deti coin"**, using many different search algorithms and type of instructions.

This coin is a file with exactly 52 bytes whose MD5 message-digest1, when printed in hexadecimal, ends with at least 8 hexadecimal zeros (i.e., its last 32 bits are all 0). The file contents must begin with "DETI coin " (note the space at the end) and must end with a newline ('\n' in C). The other bytes may have arbitrary values, but it is strongly recommended that these other bytes encode utf-8 text.

Having this in mind, we implemented the following suggested search methods:

- **AVX** - We used the AVX intrinsics to implement the search function.
- **AVX2** - We used the AVX2 intrinsics to implement the search function.
- **OpenMP** - We used OpenMP to parallelize the AVX2 search function as is the fastest one.
- **CUDA** - We used CUDA instructions to implement the search function.

# 2 Method

## 2.1 AVX and AVX2

For AVX part of the code was already given to us, we just had to implement the search function. We used the AVX intrinsics to implement the search function. The AVX2 part was implemented by us, we used the AVX2 intrinsics to implement the search function.

## 2.2 OpenMP

For OpenMP we used the AVX2 search function and parallelized it using OpenMP. We used the pragma **#pragma omp parallel for** to parallelize the search function.

## 2.3 CUDA

For CUDA we used the AVX2 search function and parallelized it using CUDA. We used the **cudaMalloc**, **cudaMemcpy**, **cudaFree** and **cudaMemcpyDeviceToHost** functions to allocate memory in the GPU, copy the data to the GPU, free the memory in the GPU and copy the data back to the CPU, respectively. We also used the **cudaMemcpyDeviceToDevice** function to copy the data from the CPU to the GPU.

# 3 Results

| **Method** | **Time (m)** | **N of attempts** | **coins found** |
|---|---|---|---|
| AVX | 2 | 1 | 1 |
| AVX2 | 2 | 1 | 1 |
| OpenMP | 2 | 1 | 1 |
| CUDA | 2 | 1 | 1 |

Table 1: Results of the experiments on processor/gpu A

| **Method** | **Time (m)** | **N of attempts** | **coins found** |
|---|---|---|---|
| AVX | 2 | 1 | 1 |
| AVX2 | 2 | 1 | 1 |
| OpenMP | 2 | 1 | 1 |
| CUDA | 2 | 1 | 1 |

Table 2: Results of the experiments on processor/gpu B

# 4 Conclusions

CUDA>OpenMP>AVX2>AVX