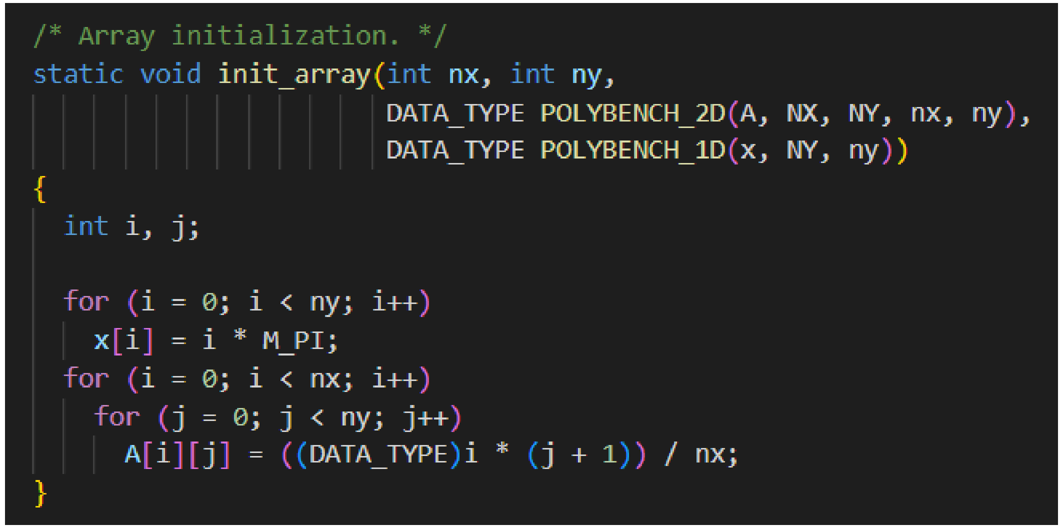
HPC ASSIGNMENT 2 – PARALLELIZING CUDA

GROUP 3 – STEFANO PIGOZZI, CATERINA GAZZOTTI AND FABIO ZANICHELLI

Optimization on initialization

ORIGINAL SEQUENTIAL CODE



```c
/* Array initialization. */
static void init_array(int nx, int ny,
                       DATA_TYPE POLYBENCH_2D(A, NX, NY, nx, ny),
                       DATA_TYPE POLYBENCH_1D(x, NY, ny))
{
  int i, j;

  for (i = 0; i < ny; i++)
    x[i] = i * M_PI;
  for (i = 0; i < nx; i++)
    for (j = 0; j < ny; j++)
      A[i][j] = ((DATA_TYPE)i * (j + 1)) / nx;
}
```

OPTIMIZED CODE ON DEVICE

Optimization on `kernel_atax`

ORIGINAL SEQUENTIAL CODE

OPTIMIZED CODE ON DEVICE

What’s the speedup?

- The optimized version takes 0,538 seconds to execute all the program
- So, the speedup is $\frac{old\ time}{new\ time} = \frac{0{,}278}{1{,}89} = 6{,}79$
- Where do we achieve this speedup?

Profiling

- Is it better to optimize the initialization or the `kernel_atax` function?
  - Obviously both

- Blablabla su dove c’è più speedup blablabla

Experiments on other datasets

	Mini dataset	Small dataset	Standard dataset	Large dataset	Extralarge dataset
Sequential times	$1{,}33 * 10^{-5}$s	0,00751s	0,419s	1,89s	4,42s
Optimized times	$1{,}77 * 10^{-3}$s	0,0108s	0,0647s	0,278s	0,665s
Speedup	**0,0075**	**0,69**	**6,48**	**6,79**	**6,64**

*Speedups written in red are slowdowns

OpenMP **vs** CUDA

	Speedup with OpenMP	**Speedup with CUDA**
Mini dataset	**0,476**	0,0075
Small dataset	**3,116**	0,69
Standard dataset	3,273	**6,48**
Large dataset	3,53	**6,79**

For mini dataset there is no best option than sequential code; OpenMP has only less slowdown

Thanks for the attention!