

# TP 7 : C++ parallel algorithms

December 13, 2023

## Objective

The goal of this assignment is to parallelize the Heat equation code using standard library parallel algorithms in C++. Students will test their code on a multicore CPU on the Baobab cluster, and on a GPU. The assignment also includes reporting the execution time ratio between CPU and GPU implementations.

## Instructions

1. Choose a domain size of  $10^4 \times 10^4$  for the 2D domain.
2. Use the `std::for_each` algorithm with the `par_unseq` execution policy for parallelization.
3. Store the 2D domain in a linear `vector<double>`.
4. Test the code on a multicore CPU on the Baobab cluster with the `cpus-per-task` variable set to 32 in the SLURM SBATCH file.
5. For successful compilation and running, include the following commands in the SLURM SBATCH file:

```
module load foss/2020b
module load tbb
```

6. To test the code on GPU, use Baobab with `public-gpu` in the SBATCH file. Include the following commands for compilation:

```
module load foss/2020b
module load CUDA
```

7. Report the ratio of execution times between CPU and GPU implementations.

8. Provide a detailed report, code, and a Makefile with commands to compile on both CPU and GPU.
9. Refer to the slides for Week 8 for additional information on std parallelism.

## Report Format

The report should include the following sections:

1. Introduction: Briefly explain the problem, the parallelization approach, and the choice of execution policies.
2. Methodology: Describe the implementation details, including the use of `std::for_each` and the storage of the 2D domain in a linear vector.
3. Results: Report the execution time ratio between CPU and GPU implementations.
4. Discussion: Analyze the performance and discuss any challenges faced during the parallelization process.
5. Conclusion: Summarize the key findings.