



NTNU

TDT4200 - PARALLEL COMPUTING

---

# Problem set 7

*Mathias Ose*

---

November 10, 2015

## MPI

The creation of the four blurred images is parallelized as the comments suggest. The rank 0 process performs the tiny blurring and receives the images from the three other processes, then performs finalization.

In my implementation each process reads the file for itself. It would also be possible for one process to read then broadcast, but I decided against that since that would just be replacing one kind of I/O with another kind of I/O, and the solution seemed fast enough.

## OpenMP

The iteration over the y-axis in the blurring loop is divided into equal parts which are parallelized with an OMP directive. Some modifications are done to the loop body so that the parts may "peek" at pixels which are just outside the "domain" of that particular thread.

## CUDA

For the functions that convert PPM to accurate format, perform blurring and perform finalization the same steps are applied: The per pixel loop body of the reference implementation is moved to a kernel function, and the loop itself is replaced by CUDA setup, transfer to device, kernel execution, transfer from device.

## Numbers

```
$ time mpirun -n 4 ./newImageIdeaMPI
real    0m3.061s
user    0m1.915s
sys     0m0.106s
```

```
$ time ./newImageIdeaGPU 1
real    0m3.119s
user    0m1.003s
sys     0m0.333s
```

CMB	Time	Energy	EDP
Mathiabo	1.44	6.48	9.33