

# TDT4200 assignment 3

September 27, 2019

## Introduction

This document aims to answer the tasks given by the third assignment in TDT4200. The associated source code can be found at [https://github.com/eldarht/TDT4200\\_assignment3](https://github.com/eldarht/TDT4200_assignment3)

## Task 0 And comparison

### 3

The time it took for the application to complete with laplacian1Kernel on average after 5 tries on my computer (Dell LATITUDE E5450) as measured by the time command:

Iterations	Real time	scatterv/gatherv (proc: 8)	scatterv/gatherv with ghosts (proc: 8)
1	0.253s	0.743s	0.738s
1024	1m 38.798s	50.176s	54.662s

The cpu is a Intel(R) Core(TM) i5-5300U CPU @ 2.30GHz with 4 cores and two threads per core(Hyper threaded).

## Task 2

### b)

The amount of communication varies depending on the number of ranks. For this task I assume 8 ranks, as that is a good fit for a four core CPU with hyper threading.

**Before the loop (setup)**

call	description	data
MPI_Bcast	The dimensions of each image section is sendt. This will send one message to each rank, but the task of sending is delegated.	9 unsigned int
MPI_Scatterv	The whole image is divided and border duplicates	5350400 unsigned char

**The loop (Kernel convolutions)**

call	description	data
MPI_Send (even)	After each loop the even numbered ranks sends one row for each ghostborder. That is one row for rank zero and two rows for rank 2, 4 and 6. With three itterations this happen three times for each ranks borders.	53760 unsigned char
MPI_Recv (even)	After each loop the even numbered ranks receive one row for each of their borders. That is the same as for the even sends.	53760 unsigned char
MPI_Send (odd)	After each loop the odd numbered ranks sends one row for each ghostborder. That is one row for rank 7 and two rows for rank 1, 3 and 5. This also happens 3 times for each ranks borders.	53760 unsigned char
MPI_Recv (odd)	After each loop the even numbered ranks receive one row for each of their borders. That is the same as for the odd sends.	53760 unsigned char

In this implementation, the border at the end of the section will always be sendt before the border before the section.

**After the loop (Reconstruction)**

call	description	data
MPI_Gatherv	The whole image reconstructed without ghost-borders	5242880 unsigned char

In Total There are 42 sends/reviecvcs in the loop and 45 in total, if you consider scatter, gather and broadcast as one each. The total data is 9 unsigned int and 10700800 unsigned char.

**d)**

I wanted as much of the memory managment to be done by MPI as possible. Having the receive write to the ghost border that was in the bmp image channel, so as to not have to use memcpy or reconstruct the image section. I did not want to touch the *applykernel()* function, as i knew that it already worked

on the rawdata. I had read the ghost\_cell\_patterns.pdf and followed the simplest instructions for preventing deadlock. Hence the exchange/recieve order is dependent on the process rank.