

# CS546 Project

## Design Report

### Implementation details for two dimensional convolution using MPI programming techniques

This project entails implementing two dimensional convolutions on parallel computers using different parallelization techniques and models. The objectives are to design and implement parallel programs using different models for parallelization and communication. 2-D convolution can be implemented by first taking 2-D Fast Fourier Transform of each input image, then perform point-wise multiplication of the intermediate results from the 2D FFTs, followed by an inverse 2-D FFT. The tasks are:

$A = \text{2D-FFT}(im1)$

$B = \text{2D-FFT}(im2)$

$C = \text{MM\_Point}(A, B)$

$D = \text{inverse-2DFFT}(C)$

2D-FFT is performed using 1DFFT routines which was provided

To implement 2d convolution using SPMD model, MPI send and receive operations, MPI\_Send and MPI\_receive functions are used. It works in the following way, first the process A decides a message to be sent to B. Process A then sends all the data in the buffer for process B. MPI allows senders and receivers to specify message IDs with the message also known as tag.

To implement 2d convolutions using SPMD model but using MPI collective communication function, it is done in the following way, MPI\_Bcast function is used. Broadcast is one of the standard collective communication techniques. During broadcast, one process sends all the data to all the processors in the communicator. One of the main uses of broadcasting is to send out user input to a parallel program, or send out configuration parameters to all processes. Here, I have used MPI broadcast with MPI send and receive functions. Here, the main process sends the data to all the processes and all the other processes receive the data from the main process.

To implement 2d convolution using task and data parallel model, the processing units are divided into 4 groups and each one of them will be charge of one task. Here, we are using 8 processors, therefore each will have 2. Each group of processors has been grouped with MPI\_Group function.

MPI\_Wtime () function is used to find the total elapsed time. One function is used in the beginning and another is used in the end, and the difference is computed to calculate the total elapsed time.

In my code, I tried to compile and run with various inputs. It always gave the correct output. I tried giving input with varied range and it always gave the expected output. Even if there was invalid input, the code execution terminates. Therefore, the program is correct.