Parallelising a CNN using 7-nested loops

Course Project - Principles of Parallel Computing Architectures

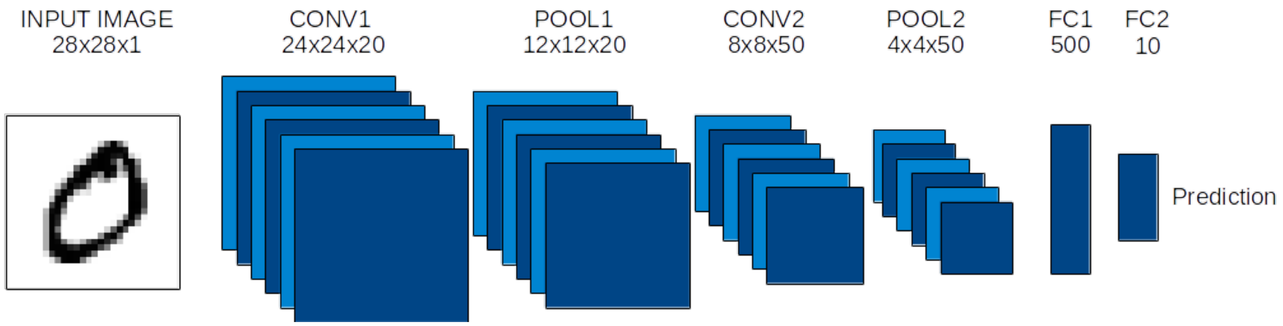
Naveen Bharadwaj

Introduction

- Convolutional Neural Networks (CNN’s).
- Implementing model parallelism using openMP, data parallelism using MPI.
- To achieve a speedup of XX over sequential execution.
- Performed both strong-scaling and weak-scaling.
- Implemented feed-forward as opposed to training on the accelerator.

Methodology and Implementation

- Have used a modified version of Le-Net 5, network architecture provided by Caffe.
- Used the MNIST hand written digit dataset, each image is a 28x28 grayscale image.
- Used ReLU activation function, Sigmoid for FC.



Methodology and Implementation

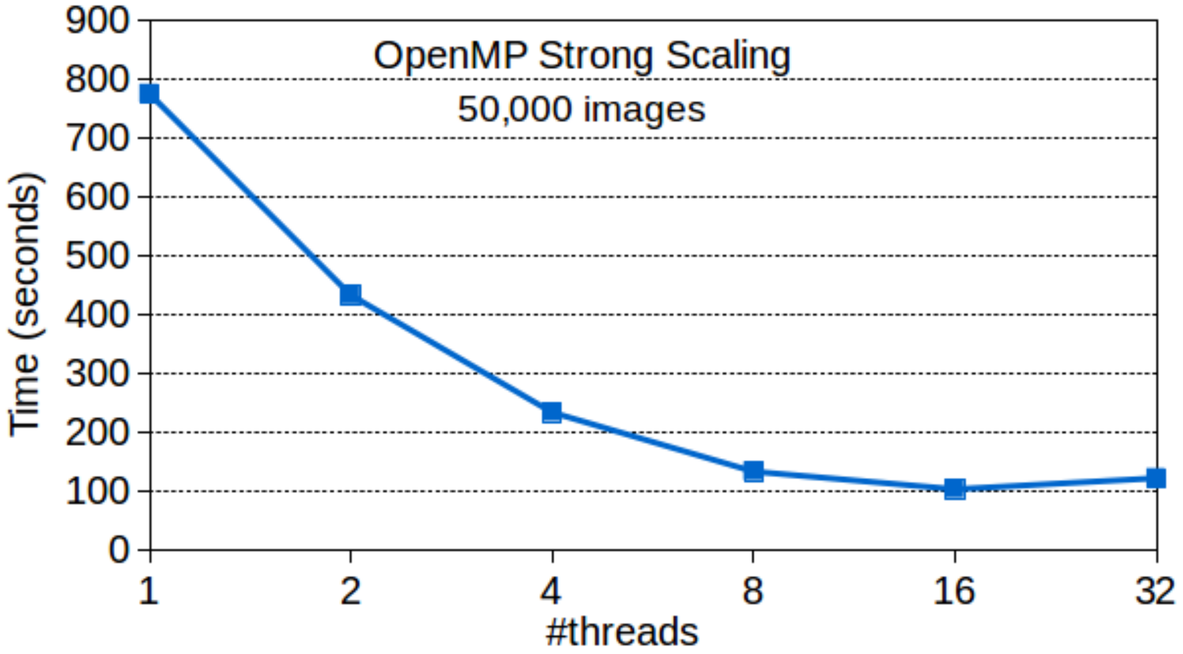
- Parallelising kernels gives us a speedup of 2 times.
- Initially planned on parallelising both the kernels and channels.
- Parallelising both kernels & channels makes the algorithm perform worse, owing to false sharing.
- Used OpenMP to parallelise the layers, MPI for parallel execution of multiple images.

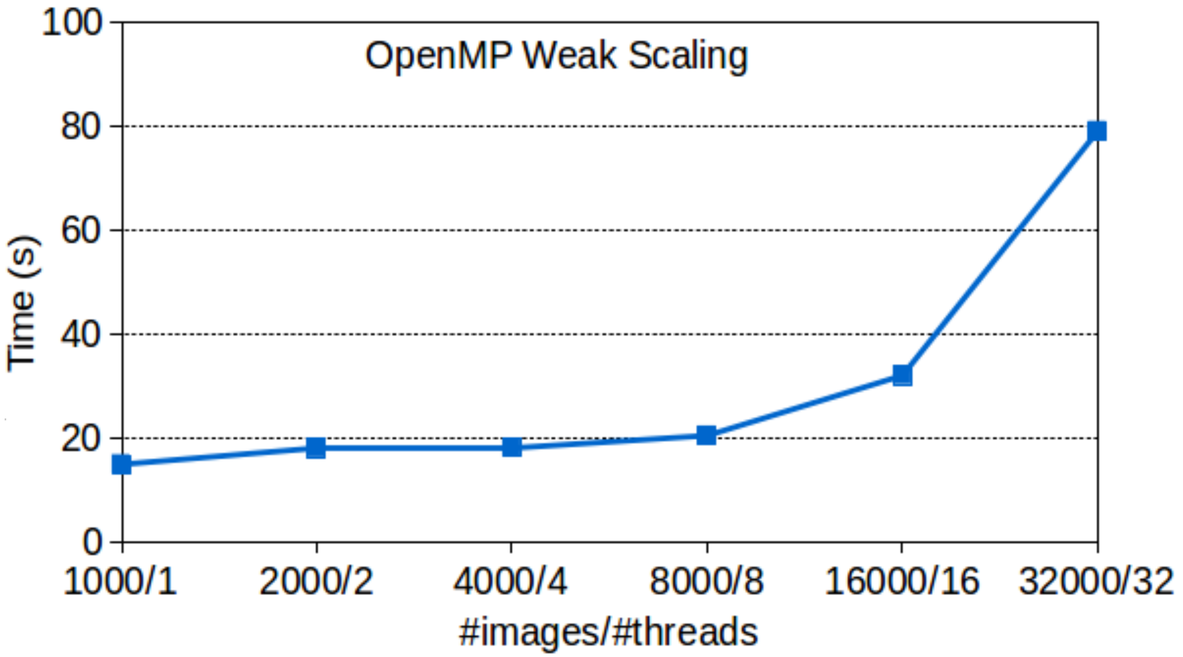
```
M - output neuron, N - Input neuron, W - Weight
for each image do                          <-- Data Parallelism - MPI
    for each layer(l) do
        for each kernel(k) do              <-- Model Parallelism - OpenMP
            for each input-row(a) do
                for each input-column(b) do
                    for each kernel-row(x) do
                        for each kernel-column(y) do
                            for each channel(c) do
                                M=N*W
                            endfor
                        endfor
                    endfor
                endfor
            endfor
            M=0
        endfor
    endfor
endfor
```

	Sequential	**Parallel K**	**K, C**	**K, C, K_y**	**K, C, K_y, K_x**
Tine (sec)	135.3	59.6	432.7	145.4	75.2

Table 1: Exec time in executing the algorithm with different levels of parallelizm; K= Kernels, C= Channels, K_x, K_y= Kernel x and y dimensions

Results

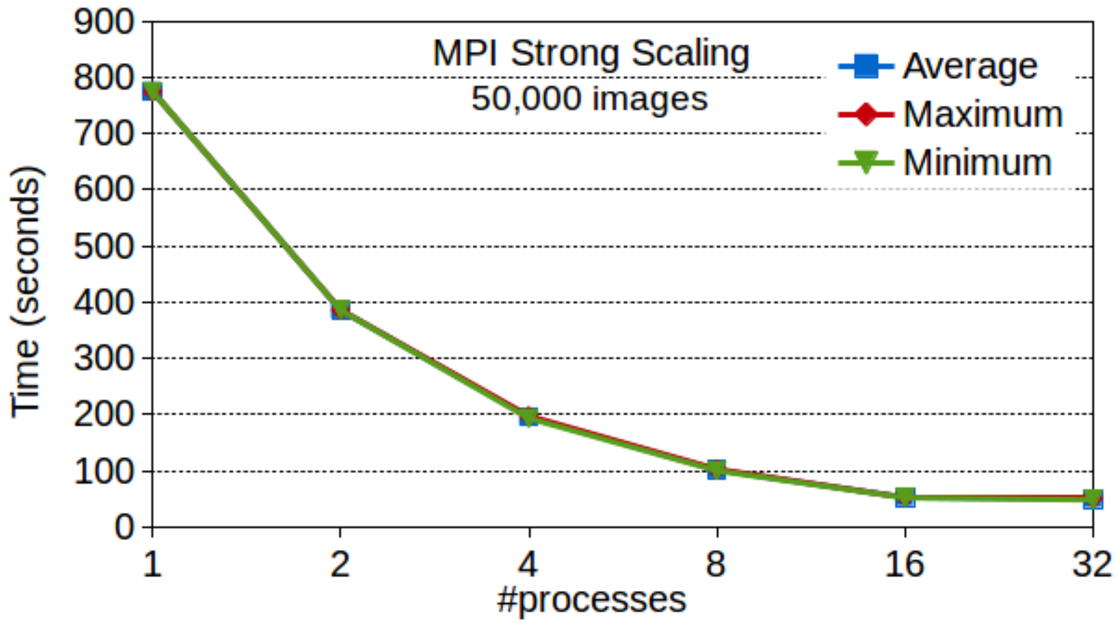


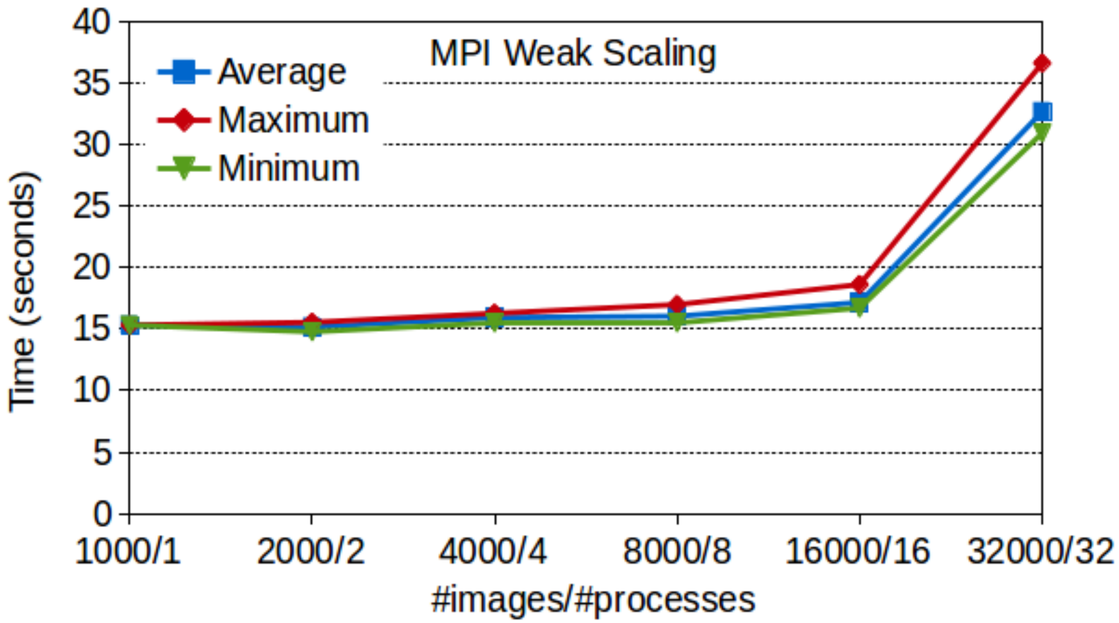


7.5x faster

Note: Results are corresponding to a single node. (Strong Scaling)

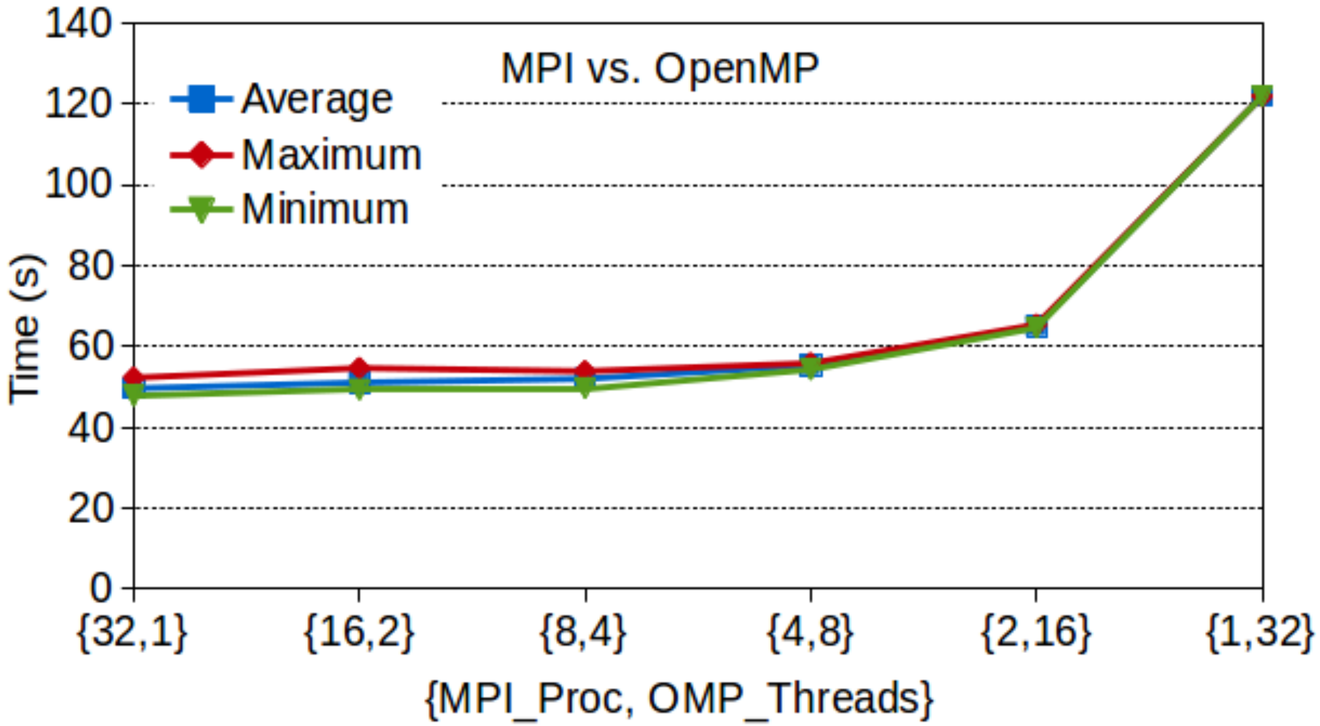
Results





2x execution time

Sensitivity Study (OpenMP v. MPI)



Using MPI over openMP has significant improvement in performance.

Thank you.

Course Project - Principles of Parallel Computing Architectures

Naveen Bharadwaj