

At first we initialised a 2D array for every process with random values and also 4 arrays to keep data communicated from all 4 directions. Then we did computation and communication of boundary values of the 2D array in direct method in every timestep. We did the same thing for packed and vector method for communication as well.

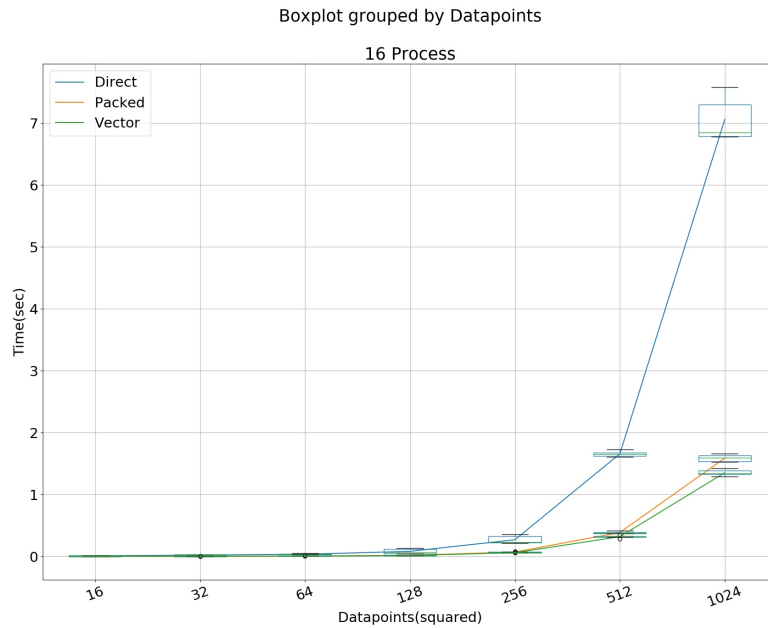
OUTPUT FORMAT:

Upon executing run.py, 4 data files(result*.txt) and 4 plots(plot*.jpg) will be generated for 16,36,49,64 processes each.

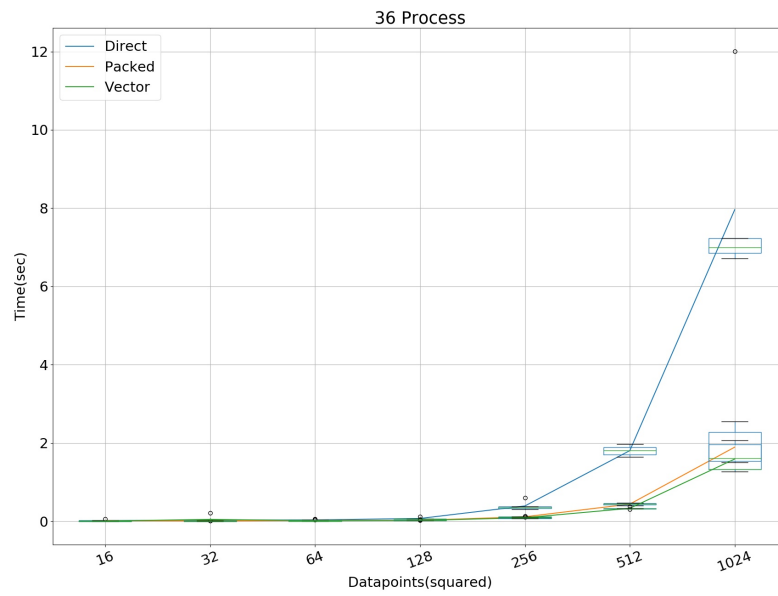
Each data file have time related to Direct, Packed and Vector methods for datapoints size 16^2 , 32^2 , 64^2 , 128^2 , 256^2 , 512^2 , 1024^2 each executed 5 times.

There is also a file called "output.txt" made with the data from data files just to make the plots.

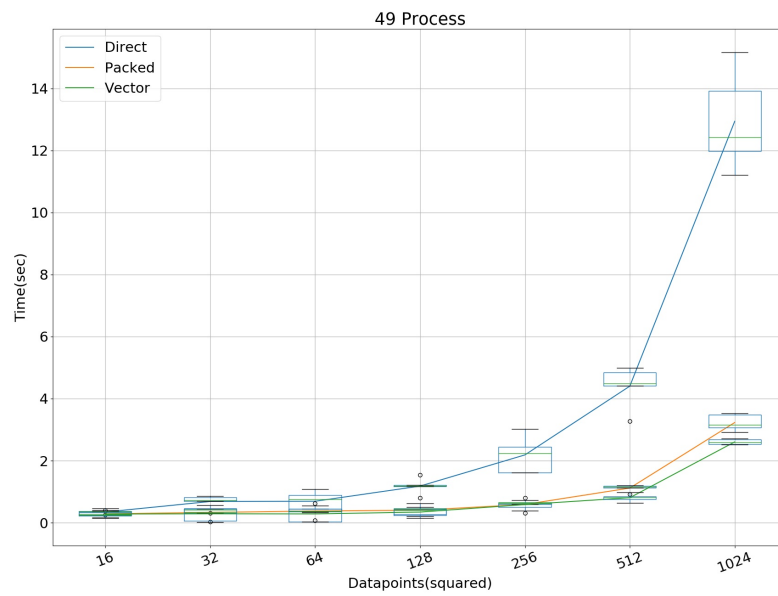
PLOTS:

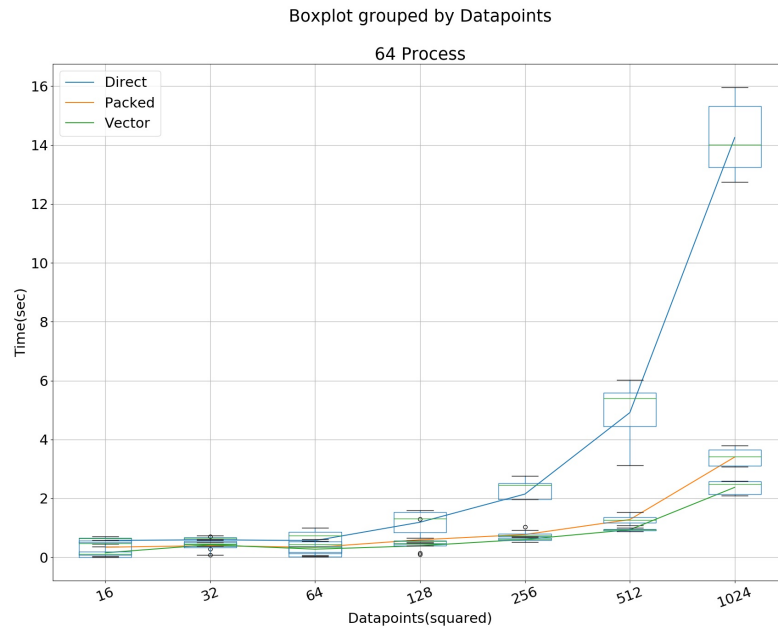


Boxplot grouped by Datapoints



Boxplot grouped by Datapoints





OBSERVATIONS:

- 1) From the plots we can see that for any number of processes, time incurred by the direct method increases drastically compared to that of packed or vector methods as we increase the number of datapoints.
- 2) In plots for all processes, we can see that the packed version takes much time compared to that of vector version as we increase the number of datapoints.

Code Execution:

python3 run.py
 plot.py script will be called internally by the run.py script, no need to execute the plot.py separately.