Problem set 3

Peter Rotar

October 2023

Question 1: As the name suggests, *non-blocking communication* allows process to not be blocked until data is sent or received, meaning, the process can continue with executing other tasks in the meantime. One example where *non-blocking communication* can be useful is when you have rank that receives data from all the other ranks. The sending ranks can in the meantime continue with tasks independent of executed one, meaning they do not intrude into sending buffer attached into MPI. Receiving rank gathers the sent data and processes it. Before we would send another data package, we can (and must) check if the previous data was received, i.e. if the buffer can be reused.

Question 2: One of the limitations of Hockey's model is, it assumes that the cost of sending message from A to B is equal to the cost of sending message from B to A. In some cases in the real world, this assumption would give false results.

Question 3: For speedup we use $S(p) = T_s/T_p$ and for efficiency $E(p) = S(p)/p$.

# of processors	time (s) (average)	speedup (average)	efficiency (average)
1	53.734686	–	–
2	37.103064	1.448254	0.724127
4	19.003608	2.827604	0.706901
8	17.598681	3.053335	0.381666

I expected that the speedup will be greater than the speedup comparison from Problem set 2, but it turns out to be roughly equal. The most likely reason is the amount of border exchange each rank has to do. You have to take into account – for some cases, that the program takes the whole boundary of a sub-grid. It can also be because I have not implemented vector type for border exchange, thus making it not so efficient.