N = 200

time\_final = 50.0

USING 1 PROCESSES

ELAPSED TIME IN MAIN LOOP = 1.859373

USING 2 PROCESSES

ELAPSED TIME IN MAIN LOOP = 1.028263

USING 4 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.654916

USING 8 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.567754

USING 16 PROCESSES

ELAPSED TIME IN MAIN LOOP = 1.708925

USING 32 PROCESSES

ELAPSED TIME IN MAIN LOOP = 2.893384

4 and 8 take the most advantage of having the work size be 50. For all the others having a work size of 50 seems to have a big impact on the time.

N = 10000

time\_final = 0.01

ELAPSED TIME IN MAIN LOOP = 0.905421

USING 2 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.460936

USING 4 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.231822

USING 8 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.121762

USING 16 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.066177

USING 32 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.036684

Having a large N with a small work size seems to do the trick. I think this is the case because the nodes don’t need to spend a lot of time on one process.

N = 1000

time\_final = 1

ELAPSED TIME IN MAIN LOOP = 0.893904

USING 2 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.461308

USING 4 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.238336

USING 8 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.135944

USING 16 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.191023

USING 32 PROCESSES

ELAPSED TIME IN MAIN LOOP = 0.184225

This last result tells me that it is better to have a small work size then more processes you have.