

TDDC78: Lab Report

Lab 5: Particle Simulation with MPI

Name	PIN	Email
Alexander Yngve	930320-6651	aleyn573@student.liu.se
Pål Kastman	851212-7575	palka285@student.liu.se

Contents

1	Introduction	3
2	Our implementation	3
3	Results	3
4	Discussion	4

1 Introduction

In this lab we were to do a particle simulation and verify the gas law $pV = nRT$ using the MPI framework.

2 Our implementation

We split the area in regions along the vertical axis, just as in the first lab. We then iterate over the amount of time.

In each time step the following is performed:

1. For all particles:
 - (a) The particle is checked for collisions.
 - (b) If a collision occurs, the affected particles are moved in new directions. Otherwise the particle is moved in its current trajectory.
 - (c) The particle is checked if it should be transferred up or down to another process.
 - (d) The particle is checked for collisions against the walls. If there is a wall collision the particles momentum is added to the processes local momentum and the particle is moved.
2. Particles are sent upwards.
3. Particles are recieved from below.
4. Particles are sent downwards.
5. Particles are recieved from above.

The total momentum is then accumulated with MPIReduce and thereafter the pressure can be calculated.

3 Results

The results can be seen in table 1 and figure 1.

Table 1 – Results from exectuions on Triolith.

Cores	Pressure	Time
1	1.7690	5.8408
2	1.5407	2.0158
4	1.3983	0.9994
8	1.2400	0.7063
16	1.1500	0.5550

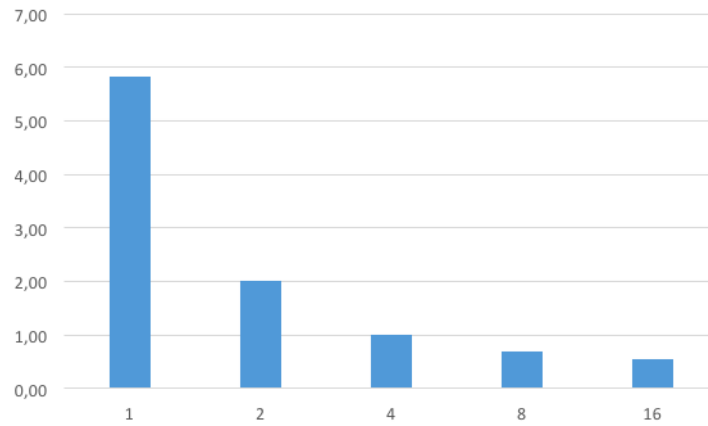


Figure 1 – Execution times for different number of cores.

4 Discussion

As can be seen in table 1 the pressure changes with the amount of cores. This is of course not correct. This is due to the way the parallelisation is implemented. If two particles collide across a border between two processes, the collision won't be detected. They will just pass through each other and be transferred to the other process. With an increasing number of cores, there will be more borders that this can happen on, which makes the pressure deviate more from the correct value (which is the value obtained using one core).