Universidad de Investigación de Tecnología
Experimental Yachay Tech

School of Mathematics and Computer sciences

Program : Undergraduate ———— Semester : VII
Subject : HPC Professor : Saravana Prakash T
Date : 23/05/2019 Time:
Semester: Jan-May-2019
Student's Surname, Name : Orozco Andrés
Student Reg.No:

Physics Simulation Project

1. Introduction

For this project we are going to parallelize particle simulation with OMP and MPI libraries. I chosed particle simulation since it is easy to simulate and can be working very easy with different integrators methods. We work with an interval time *(dt)* with it time we compute the possition and the velocity of the particle. The objective of this project is to compare how much time take the simulation when it is parallelized compare with the serial code using a different number of iterations.

2. Pseudo Code

Here are listened the pseudo codes for the chosen methods.

Runge-kutta 2

```
RK2 (position, velocity, dt, mass,T, C)
{
   k1 = dt *( 1.0f/mass * ( T - C * velocity) );
   k2 = dt *( 1.0f/mass * ( T - C * (velocity + k1 /2.0f)) );

  velocity = velocity + ( 0.5 * k1 + 0.5 * k2 ) ;
  position = dt * velocity + position;
}
```

Runge-kutta 4

```
RK4 (position, velocity,  dt, mass, T, C)
{
   k1 = dt *( 1.0f/mass * ( T - C * velocity) );
   k2 = dt *( 1.0f/mass * ( T - C * (velocity + k1 /2.0f)) );
   k3 = dt *( 1.0f/mass * ( T - C * (velocity + k2 /2.0f)) );
     k4 = dt *( 1.0f/mass * ( T - C * (velocity + k3 )) );

  velocity = velocity + (1.0/6.0f) * ( k1 + 2 * k2 + 2 * k3 + k4 );
  position = dt * velocity + position;
}
```

Euler

```
Euler (position, velocity, dt,  mass, force)
{
  velocity = dt * (mass/force) + velocity;
  position = dt * velocity + position;
}
```

3. Results

For the results we decided to increase the number of steps (interations) and compare them vs the execution time, increasing the number of steps means that the particle traveled much more. For MPI and OMP we ran the the codes in a machine with 4 processors.

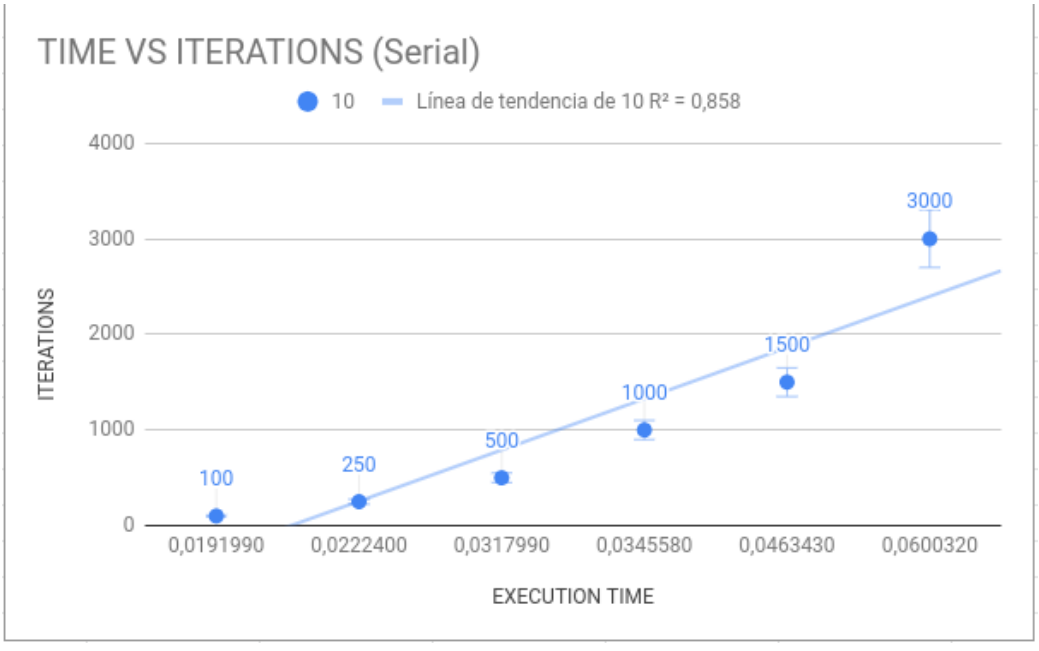

Figura 1: Execution time of serial code

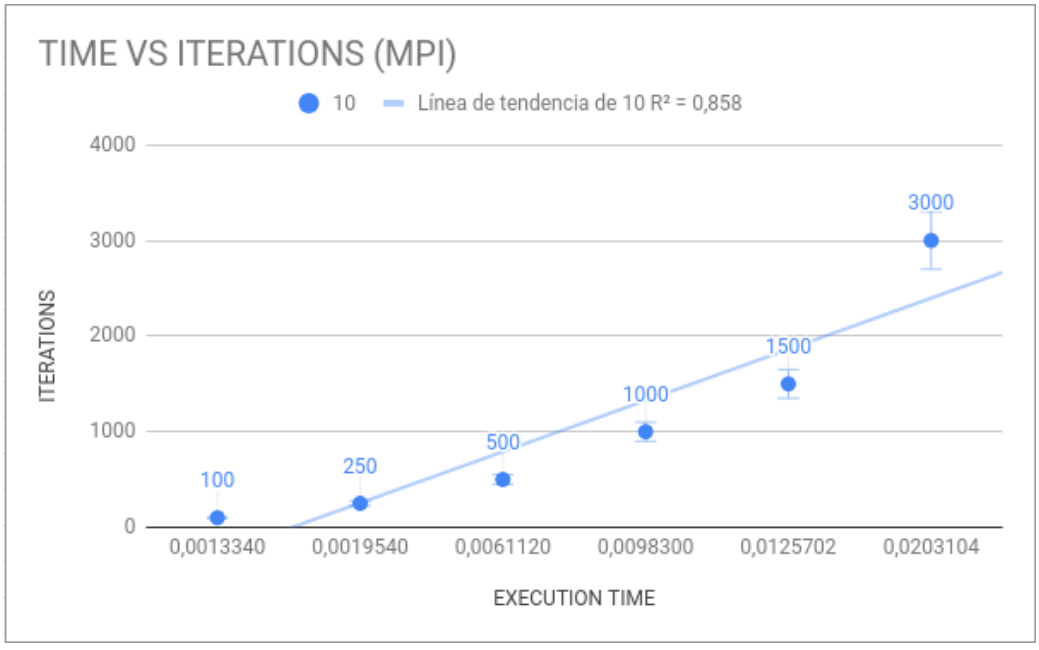

Figura 2: Execution time using MPI and 4 cores

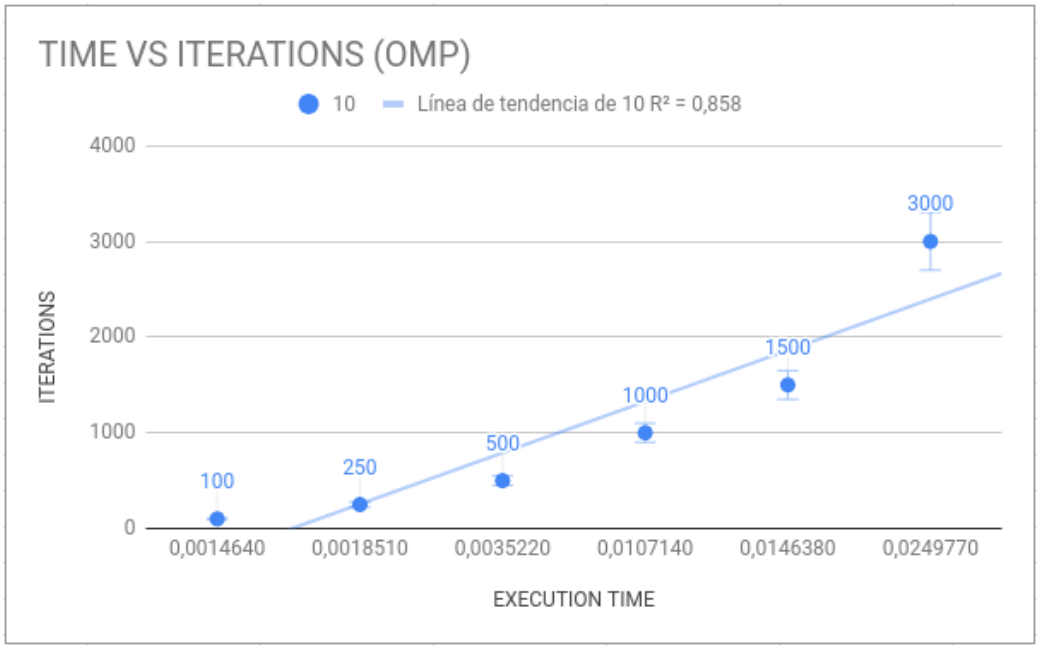

Figura 3: Execution time using OMP and 4 cores

4. Conclusions

Despite the execution times are really small for both parallel and serial we can see in the graphs the parallel codes outperformed the serial code so far, in situations of much more

complex simulations the execution time in parallel could drastically decrease, another point to emphasize is that the execution times using both OMP and MPI are very similar although OMP obtained shorter times, these times are almost imperceptible. To finish you have to remember that the simulation can be run on a computer with many more processors decreasing the execution time by a lot.

Signature of the student