CSE305 Project: N-body simulations

Gleb Pogudin, gleb.pogudin@polytechnique.edu

In this project, the task will be to simulate the dynamic of several bodies (you can think of planets, satellites, stars, etc.) moving simultaneously under the influence of gravity forces. To keep things simple, we suggest you to focus on the 2D case.

Input data consists of the description of N bodies $b_1, \ldots, b_N$, and each body is described by

- mass m_i;
- coordinates x_i, y_i;
- initial velocity u_i, v_i.

The force the bodies b_i and b_j act on each other is

$$f_{i,j} = G\frac{m_i m_j}{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad \text{where } G \text{ is the gravitatiinal constant.}$$

The program should use the Newton's laws to produce a simulation of the dynamics with a user-specified timestep.

You are expected to implement:

1. *The straightforward simulation algorithm* that, in a loop, first computes all the forces an then updates all the positions. Then try to parallelize it as follows
   (a) parallelize the update step;
   (b) avoid computing each force $f_{i,j}$ twice;
   (c) parallelize the computing of the forces, avoiding race condition using mutexes/atomic variables;
   (d) try to come up with a way of using mutexes/atomic variables as little as possible (e.g., by dividing bodies in groups to be treated by different threads separately), it is actually possible to avoid mutexes and atomic variables completely.
2. *Vizualization.* Your code should be able either produce a gif animation for the simulation (you may want to use Magick++ library[1]) or open a window with the simulation drawn in the real time (the result must be compilable on Linux, you can use gtkmm library, for example). Note that you should parallelize the computation and input-output, so that the computation is not blocked when IO happens.
3. (Optional, encouraged for teams of three)*Any other simulation algorithm*, for example, the Barnes-Hutt algorithm[2] or some other[3], and suggest ways of parallelizing it.

Your report should contain a detailed comparison of the efficiency (including the speedup depending on the number of threads) and accuracy of different versions of the code on a set of benchmarks.

[1] https://imagemagick.org/Magick++/

[2] Here is a paper https://www.nature.com/articles/324446a0, here is a friendlier description http://arborjs.org/docs/barnes-hut

[3] here is a list: http://www.cs.cmu.edu/~guyb/real-world/nbody/index.html