Parallel Computing I
Homework 8: N-body simulation

Thorsten Grahs, Andres Rodriguez

6. Juli 2015

This will be your eigth homework in the exercise parallel computing. Send your solution to a.rodriguez-escobar@tu-braunschweig.de until **July 22th 2015 23:55**.

Prepare a pdf file for your written text and attach the source code of your program to the mail.

In this task you are going to implement a 2D N-body simulation in parallel.
The general procedure could be the following

Algorithm

- **Initialization($n = 0$):**
  1. Create N celestial bodies (suns, planets, comets etc.).
  2. Initialize location (2D), velocity(2D), mass (1D), radius (1D) and type randomly.
  3. Avoid collisions. i.e bodies should not interfere each other.
- **Simulation loop ($n = 0, 1, 2, \ldots$):**
  1. Compute for each body i the gravitational force $\vec{F}_i^n$.
  2. Update positions: $\vec{x}_i^{n+1} := \vec{x}_i^n + \Delta t \cdot \vec{v}_i^n$
  3. Update velocities: $\vec{v}_i^{n+1} := \vec{v}_i^n + \frac{\Delta t}{m_i} \cdot \vec{F}_i^n$
  4. Resolve collisions
  5. Correct domain mapping

Task 1 N-body simulation (20 Points)

- Implement the N-body problem, following the algorithm above.
- Use a domain-size of 512×512 cells and bodies of different type.
  The types should differ in size/radius, mass and colour.
- Handle collisions with an back-bouncing approach
- Use a domain decomposition of at least 4 sub-domains.
- Assume cyclic boundaries. That means:
  – Particles leaving the upper domain boundary enter again at the lower boundary.
  – Particles leaving the right domain boundary enter again at the left boundary.
  And vice-versa.
- Use derived and packed MPI data-types to map leaving bodies to another domain (send via MPI to another processor)
- Conduct simulations and make a video (30-40 seconds)

Bonus task (5 points)

- Integrate a load-balancing with dynamic domain adaptation in your N-body program.

Explain in words, what your program is doing!