

## Module 5.9: Instructor Guide

- There is a Makefile in the .tar.gz file for the source code. It may need to be adapted to your system.
- This unit is a walk through parallelizing a simple N-body code using MPI. We recommend the instructor use the questions posed in the slides as discussion starters and provide the serial code to the students as required reading for the class.
- Slides 2-9 appear in the similar OpenMP and OpenACC lessons -- if more than one is being used, the slides can be omitted. Alternatively if time is an issue they can be provided as background.
- There are 3 sample assessments -- The first two simply require changing the number of bodies and rerunning across a number of processors (and of course plotting)
- The 3rd sample assessment involves a rewrite of sorts -- the IO section at the end of the code needs rewriting, the pseudo code is as follows:
  - Loop over time index
    - Gather data to process 0
    - Write data to disk
  - The serial code might prove to be a useful starting point as it is essentially this without the gather
- On slide 16, the graph shows that for 16 processors OpenMP (there is a nearly identical lesson in the OpenMP content of this project) is faster than MPI -- this is because the MPI code has extra communication overhead. This is a great example of why a hybrid approach (MPI + OpenMP) is desirable as you can obtain the best of both worlds.

### Lesson 5.9: Common Pitfalls for Students and Instructors

None known at this time, but it is expected that Computer science students may have a harder time with the physics content (Newton's laws) and potentially Verlet's method. Students from a science background may struggle more with some features of the C++ language such as the use of a macro to quickly convert what might more naturally be a 2D array in MATLAB/FORTRAN into a 1D array.