CSE: HW 3

Evan Ott
UT EID: eao466

November 21, 2016

I plan to complete the canned project, but as previously discussed, I wanted to create a stretch goal to include MPI. I've used ØMQ in Python up at ARL to create a master-worker distributed system before for basic message passing, so the concept of distributing computation across multiple nodes doesn't scare me. I just don't have experience using the more standard tool of MPI, and don't have experience running in a general-use environment like Stampede.

As for exactly what the goal is, I plan to write an additional version of the second problem in the canned project (charged particle motion in a vertical electric field) that will simulate trajectories for many (non-interacting) particles across multiple cores on more than one node. I'll probably use a master-worker solution wherein one core is responsible for sending out initial conditions $(x_0, y_0, z_0, \dot{x}_0, \dot{y}_0, \dot{z}_0)$ to the other cores.

I know I could also configure the code to read from a common input file and select the conditions based on the MPI rank field (taking indices where the modulus equals the rank, or similar), but I think it's slightly more of a challenge to manage the message passing, especially if there are more particles than total cores available for computation.

I will test the parallel computation's results against the one-off computation, to at least be sure nothing gets lost in the message passing. Of course, I'll write all the code in a modular way so that the simulation code will really be exactly the same whether it's being run using MPI or just a single core. Only the manner in which it gets called will really change.