High Performance Computing: Final Project Proposal

Paul Beckman, Mariya Savinov

March 28, 2022

We would like to implement and test the parallelization of tree code [1] / fast multipole methods [2] for the rapid evaluation of the total potential for a collection of particles interacting through a conservative force, e.g. the electrostatic potential for a collection of point charges. As a brief exposition of the problem, consider points $x_1, ..., x_n \in [0, 1] = I$ with charges $q_1, ..., q_n$. We limit the positions to a one-dimensional problem so that we can focus on computational aspects, but imagine we are embedded in $\mathbb{R}^3$ so that the potential at x of a particle at point x_j with unit charge is

$$\phi(x) = \frac{C}{|x - x_j|}. \tag{1}$$

where C is some scaling factor (e.g. the Coulomb constant) which we will take to be $C = 1$. We would like to evaluate the potential

$$u(x_i) = \sum_{j=1}^{n} q_j \phi(|x_i - x_j|) = \sum_{j=1}^{n} \frac{q_j}{|x_i - x_j|} \tag{2}$$

for each x_i, which in practice one would then use to compute forces between particles. The tree code / fast multipole idea is to hierarchically partition the interval $[0, 1]$ using a binary tree, and accurately approximate

$$u_k(x_i) = \sum_{x_j \in I_k} q_j \phi(|x_i - x_j|) \tag{3}$$

for all subintervals I_k sufficiently far from x_i by representing ϕ with its truncated Taylor series. One then repeats this computation for various levels in the hierarchy.

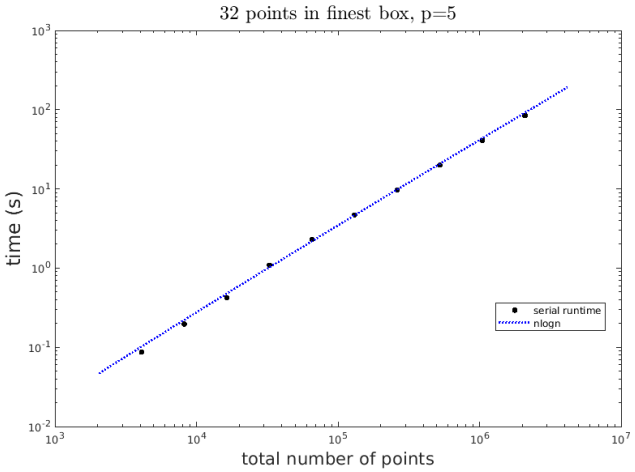
The primary computational questions involve the efficient distribution of the hierarchical data structure and computations across processes. We plan to approach these questions through the following tasks

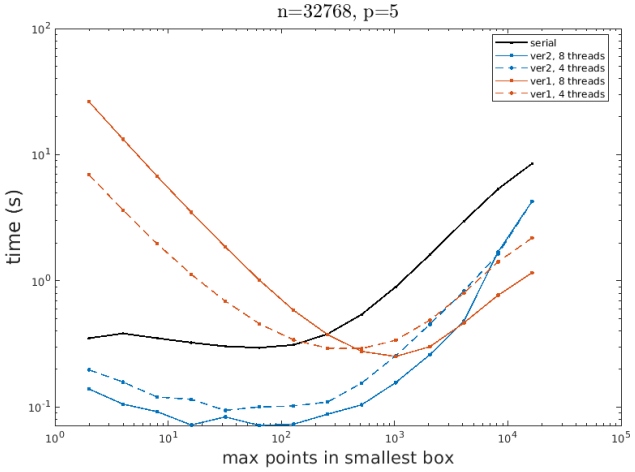
- Implement a serial $\mathcal{O}(n \log n)$ Barnes-Hut type tree-code
- Parallelize this approach using OpenMP, experimenting with tuning for improved cache usage
- Parallelize this approach using MPI, experimenting with communication minimization between cores
- (If time allows) improve tree code to an $\mathcal{O}(n)$ fast multipole method and modify parallelism accordingly

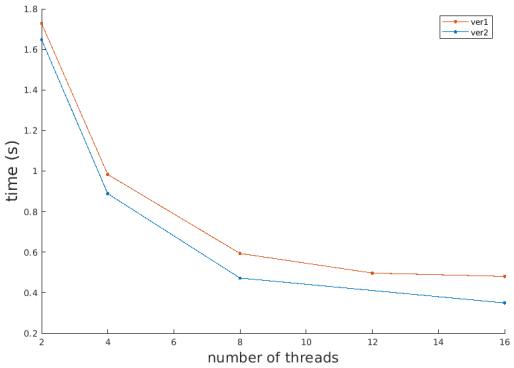
References

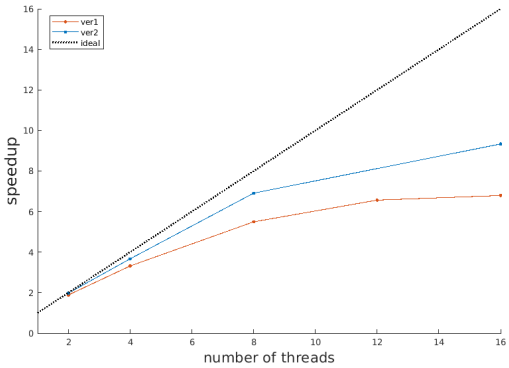
[1] Josh Barnes and Piet Hut. A hierarchical O(NlogN) force-calculation algorithm. *Nature*, 324(6096):446–449, 1986.

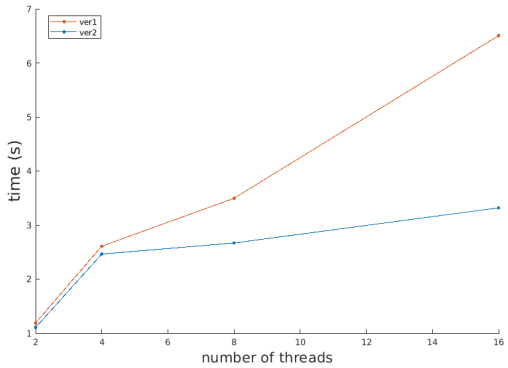
[2] Leslie Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.

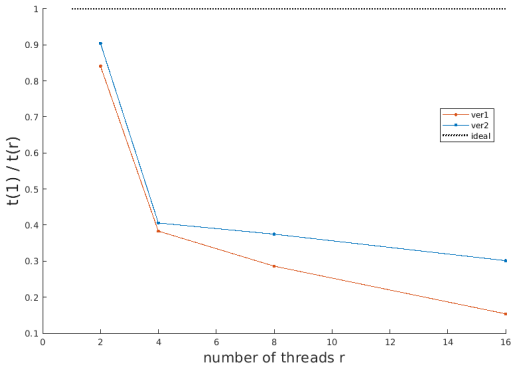

32 points in finest box, p=5
time (s)
total number of points
serial runtime
nlogn


n=32768, p=5
time (s)
max points in smallest box
serial
ver2, 8 threads
ver2, 4 threads
ver1, 8 threads
ver1, 4 threads


time (s)
number of threads
ver1
ver2


speedup
number of threads
ver1
ver2
ideal


ver1
ver2
time (s)
number of threads


ver1
ver2
ideal
t(1) / t(r)
number of threads r