COMP 633: Parallel Computing
Programming Assignment 1(b)
Parallel all-pairs n-body simulation

Assigned: Tue Sep 28
Due: Tue Oct 12 (at start of class)

In this assignment you will use OpenMP to parallelize the two sequential n-body programs constructed in PA1(a). First update your PA1(a) sequential all-pair and half-pair programs as needed for correctness and performance. Verify the resulting programs compute approximately the same results as the all-pair reference implementation. Your half-pairs solution must make use of Newton's third law, $f_{ij} = -f_{ji}$, to halve the number of force calculations needed compared to the all-pairs solution. You can use or adapt these sequential solutions, if you wish, as a starting point for this assignment.

Programming Assignment

1. Plot the performance of your all-pairs and half-pairs *sequential* n-body programs for $k = 6$ time steps with values of $n \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$ on the *x* axis and the observed interaction rate $R(n,p) = kn^2/t_p$ on the y axis (in units of millions of interactions per second). Here t_p is the wall-clock time to perform k time steps. Add the complete command line used to compile the programs on the plot or on an attached sheet.
2. Add OpenMP parallelization directives to your two n-body programs in (1) and experiment to maximize performance. Insure the final versions maintain approximately the same results using the reference implementation. Use the `omp_get_wtime()` function from OpenMP in the master thread to obtain t_p, the elapsed time in seconds.
3. Plot the parallel performance of your two programs in (2) on separate graphs. The x-axis of the graph is the number of bodies $n \in \{200, 500, 1000, 2000, 5000, 10000, 20000\}$, and the y-axis is the performance $R(n,p)$ in millions of interactions per second. For each $p \in \{1, 4, 8, 16, 18, 20\}$ plot and connect the points $(n, R(n,p))$. Add the complete command line used to compile the programs and the settings of any OMP-specific environment variables such as KMP_AFFINITY or GOMP_AFFINITY and OMP_NUM_THREADS.

Submission

Submit your code like pa1a: send me the three performance plots before the start of class on the due date and leave a copy of your programs (all-pair and half-pair) in your pa1b directory used in (3) above. You can work together with a partner, if you wish just as in pa1a (be sure to include both names on the submission).