COMP 633: Parallel Computing
Programming Assignment 1(b)
Parallel all-pairs n-body simulation
Fall 2018

Assigned: Tue Oct 2, 2018
Due: Tue Oct 16, 2018 (at start of class)

In this assignment you will use OpenMP to parallelize the two sequential n-body programs constructed in PA1(a). You are free to modify your PA1(a) programs to improve sequential performance, but they should compute (within a small epsilon) the same result as the (all-pair) reference implementation provided on the web. In addition, 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.

Programming Assignment

1. Modify your PA1(a) sequential programs as needed for correctness and optimize sequential performance while retaining the basic methods (all-pairs and half-pairs). Verify the resulting programs compute approximately the same results as the all-pair reference implementation.

2. Plot the performance of both your sequential n-body programs from (1). For values of $n \in \{10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000\}$ on the x axis, and using $k = 4$ time steps, plot performance on the y axis in units of millions of interactions per second, exactly as you did for PA1(a). You do not need to show variance on this plot, but do show the command line you used to compile the programs.

3. Add OpenMP parallelization directives to your two n-body programs from (1) and experiment to maximize performance. Insure the final versions maintain approximately the same results using the reference implementation. To measure performance you can use the interaction rate $R(n, p) = kn^2 / t_p$ in units of interactions per second, where $t_p$ is the wall-clock time to complete $k$ iterations using $p$ processors. Use the omp_get_wtime() function from OpenMP in the master thread to obtain elapsed time in seconds.

4. Plot the parallel performance of your two programs in (3) on separate graphs. The x-axis of the graph is the number of bodies $n$, 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 $(n, R(n, p))$ for $n \in \{200, 500, 1000, 2000, 5000, 10000, 20000\}$. Include 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: turn in the three performance plots at the start of class on the due date along with a listing of your parallel all-pairs and half-pairs n-body programs developed in (3) above. You can work together with a partner, if you wish (be sure to include both names on the submission).