Project #2
OpenMP: N-body Problem -- Coarse vs Fine and Static vs Dynamic

Haoxiang Wang; Student ID: 932359049

1. Machine Running the Test

I used school's server "flip" to run the test. At the time my test is running, the "uptime" is around 8.4.

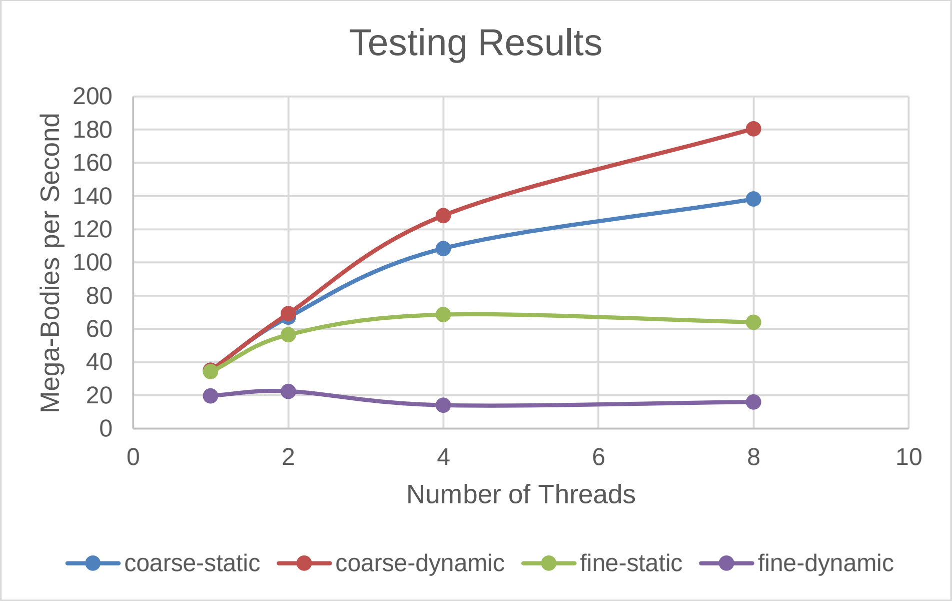
2. Performance Results

I used Benchmarks to let the program running with different number of threads. The number of thread I used for the test are 1, 2, 4 and 8. Since the problem size relates to the number of bodies, and the number of bodies is fixed to 500, the size of the problem is the square of the number of bodies which is 2500. Also, I used 500 steps to implement the calculation, so the number of steps is 500 and the total runtime covers 2500 * 500 = 1250000 times of calculation of bodies. The performance is evaluated by calculating mega-bodies computed per second. Which is dividing the total bodies by total runtime.

The tests are set to four types. They are coarse-grained parallelism with static and dynamic scheduling, and the fine-grained parallelism with static and dynamic scheduling. The following form is the result I got from the test.

	1	**2**	**4**	**8**
coarse-static	**35.14**	**67.03**	**108.46**	**138.16**
coarse-dynamic	**35.07**	**69.30**	**128.26**	**180.52**
fine-static	**34.26**	**56.53**	**68.69**	**64.11**
fine-dynamic	**19.76**	**22.39**	**14.11**	**16.06**

Results show in graph:



3. Behavior Explanation

The results shown in the graph are interesting. When the number of threads goes up, the performance of coarse-grained parallelism goes up but the fine-grained parallelism keeps even. Also, the coarse-grained parallelism performs better than fine-grained parallelism at all times. This makes sense since both of these two methods work on same size of problem and the coarse-grained parallelism creates the threads once for all while fine-grained parallelism creates threads for each outer loop. The creation of threads takes time and the problem size is not big enough to ignore the effect.

There's another thing that should be noticed in the graph, which is that under the coarse-grained parallelism, dynamic scheduling performs better than static while under fine-grained parallelism static performs better. This could be explained using overhead of the program. Since the coarse-grained parallelism only separates problems to threads for one time, the dynamic scheduling could show its efficiency which will make the result has less effect from the overhead. However the fine-grained parallelism has to separate problem to threads for each outer loop, the overhead could not be ignored during the process and this results in worse performance.