Bryce Egley
CS475

Project 2 Write Up

1. Tell what machine you ran this on

Flip (Linux) which has 24 processors

2. Create a table with your results.

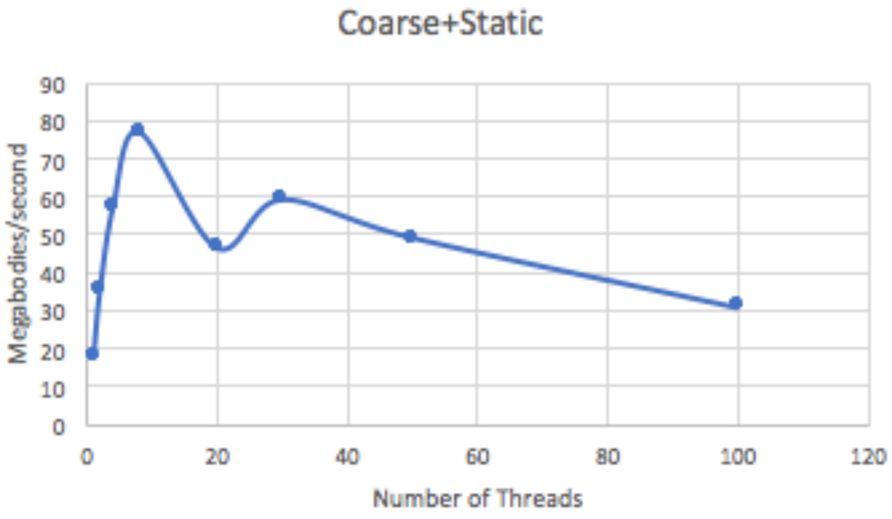
Using 100 Bodies 200 steps

NumThreads	Megabodies/Second	schedule	grain
1	17.3015	dynamic	coarse
2	34.4321	dynamic	coarse
4	53.6595	dynamic	coarse
8	81.4476	dynamic	coarse
20	41.4171	dynamic	coarse
30	56.286	dynamic	coarse
50	53.2944	dynamic	coarse
100	37.519	dynamic	coarse
1	17.9893	static	coarse
2	35.4617	static	coarse
4	57.2263	static	coarse
8	76.9138	static	coarse
20	46.6154	static	coarse
30	59.2395	static	coarse
50	49.2686	static	coarse
100	31.0105	static	coarse
1	11.8708	dynamic	fine
2	19.2832	dynamic	fine
4	9.09995	dynamic	fine
8	8.68168	dynamic	fine
20	5.58825	dynamic	fine
30	1.24179	dynamic	fine
50	0.920859	dynamic	fine
100	0.464927	dynamic	fine
1	16.3333	static	fine
2	26.821	static	fine
4	19.1316	static	fine

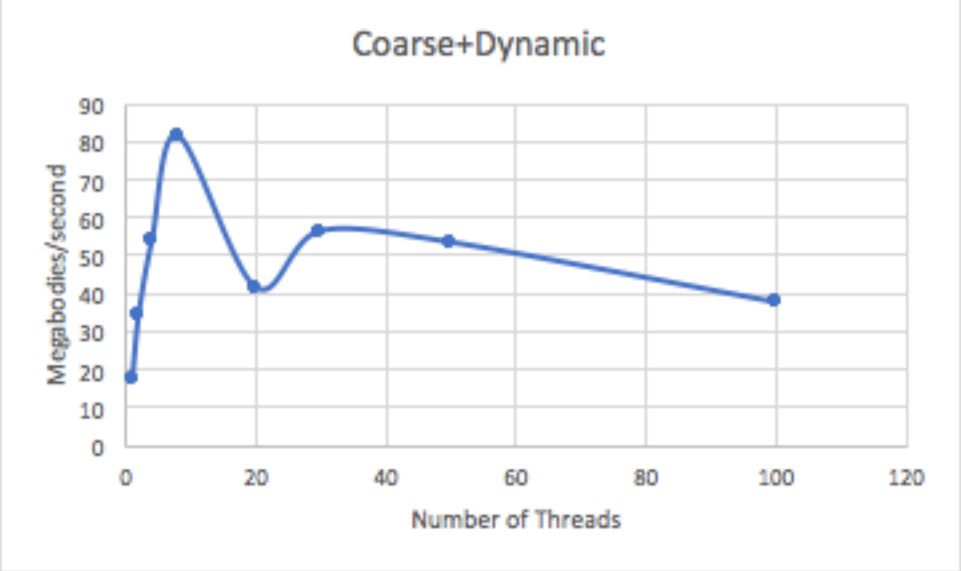
8	15.3076	static	fine
20	1.97259	static	fine
30	1.62212	static	fine
50	0.949757	static	fine
100	0.491028	static	fine

3. Draw a graph. The X axis will be the number of threads. The Y axis will be the performance in whatever units you sensibly choose. On the same graph, plot 4 curves:

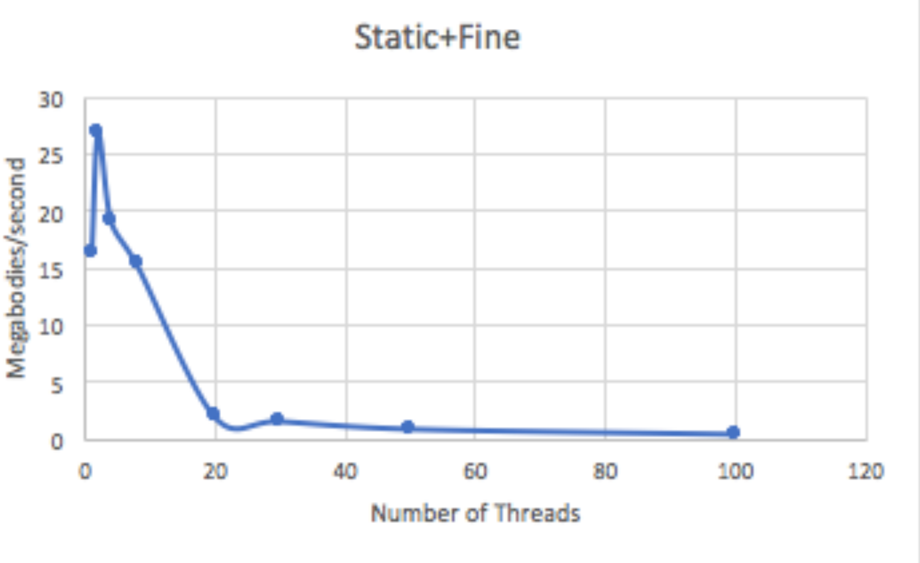
   a. coarse+static



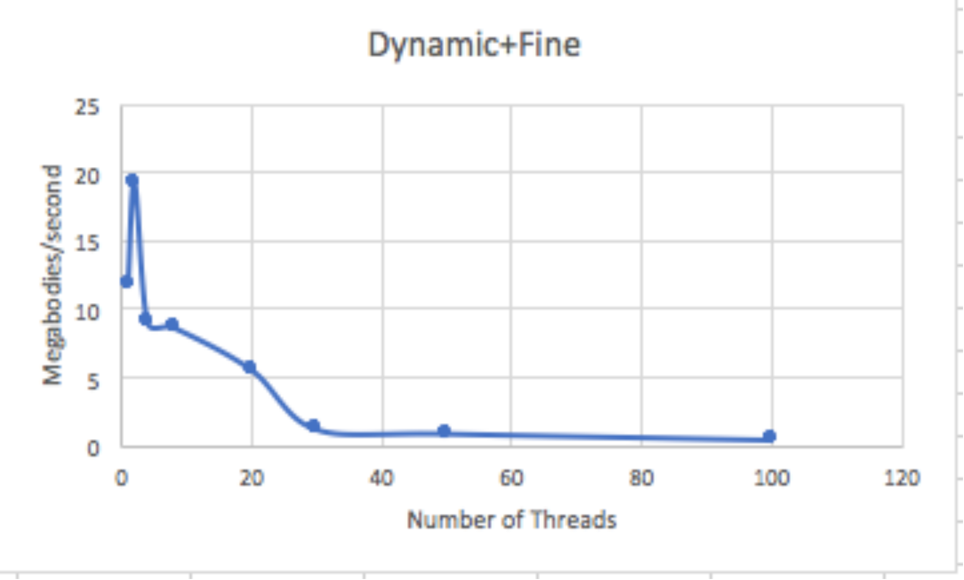
b. coarse+dynamic



c. fine+static



d. fine+dynamic



4. What patterns are you seeing in the speeds?

   I notice that all 4 graph increase initially, then decrease. This turning point in fine happens at 2 threads and in coarse happens at 8 threads.
   There doesn't seem to be huge difference between static and dynamic. Static looks slightly better. Coarse seems much better in terms of megabits/second then fine.

5. Why do you think it is behaving this way?

   The overall trend we see of increase and then decrease pattern follows the performance as a function of number of threads that we learned about in lecture. More threads divide up the work, however once you have too many threads then too much time is spent on communication between the threads and less on work.
   It seems that coarse grained works better than fine grained since the computing is being split into larger tasks whereas fine grained computing is broken into make small tasks.