

Name: Hanif Khan

CS_575 OSU Project 2

1. Tell what machine you ran this on

I ran this program on 2 different machines with the different compiler.
I ran on my virtual machine and also on the flip server.

2. Create a table with your results.

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:

1. Coarse static
2. Coarse dynamic
3. Fine static
4. Fine dynamic

Using IPCC compiler on flip server.

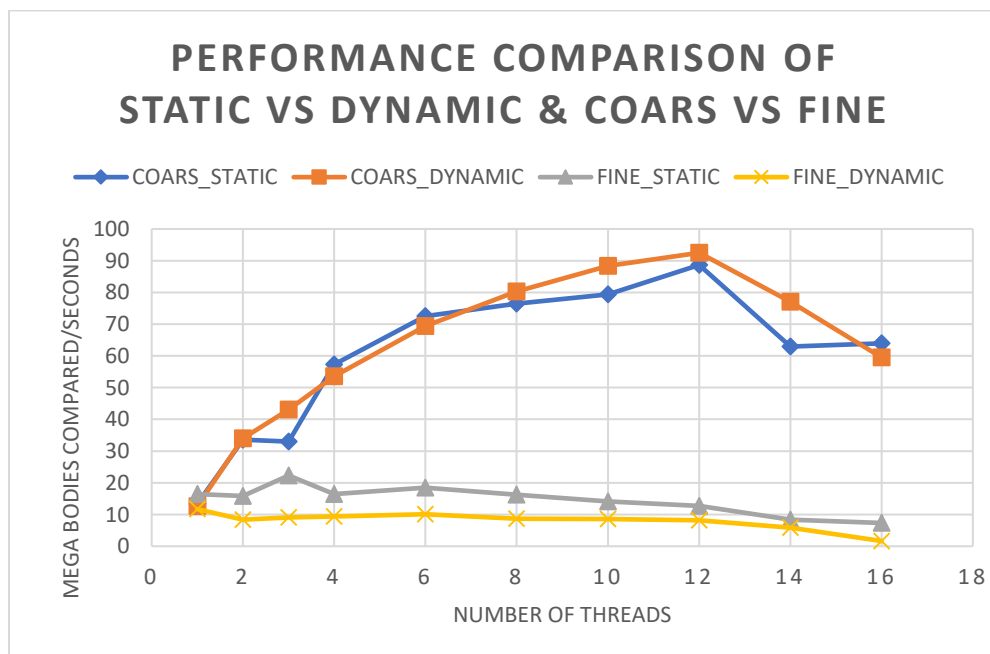
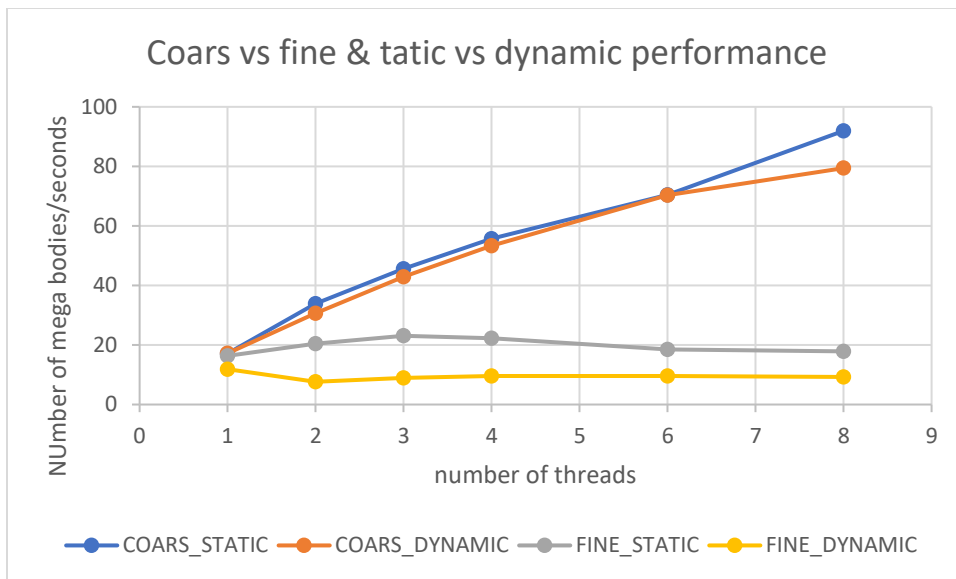


Table:

	COARS_STATIC	COARS_DYNAMIC	FINE_STATIC	FINE_DYNAMIC
--	--------------	---------------	-------------	--------------

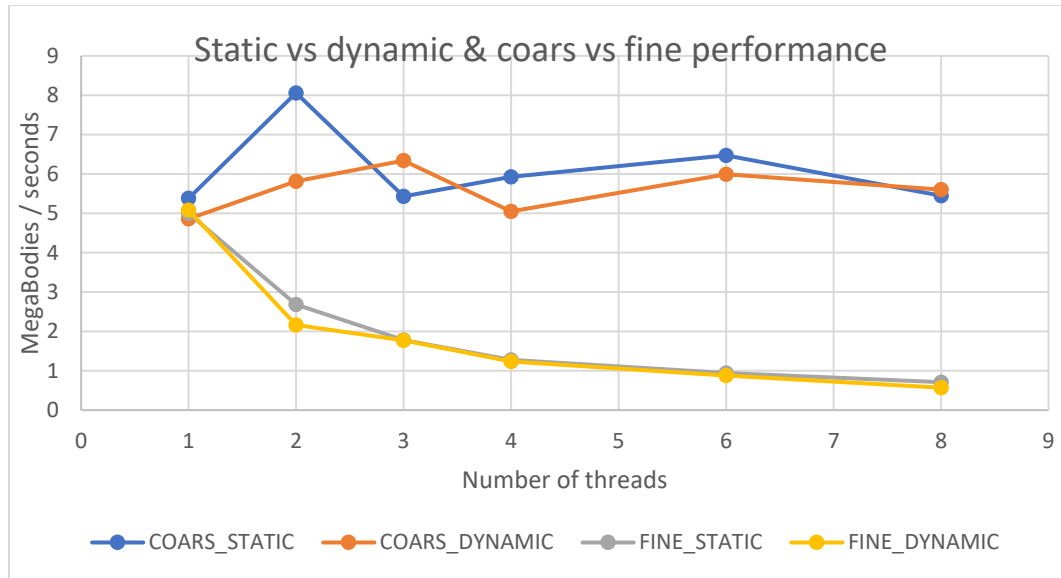
1	13.35	12.61	16.47	11.7
2	33.63	34.01	15.85	8.35
3	33.01	43.05	22.31	9.13
4	57.29	53.51	16.48	9.4
6	72.49	69.38	18.5	10.1
8	76.43	80.3	16.26	8.72
10	79.4	88.31	14.14	8.54
12	88.65	92.49	12.69	8.16
14	62.98	77.03	8.34	5.84
16	63.98	59.52	7.34	1.65

Using G++ on flip server



	COARS_STATIC	COARS_DYNAMIC	FINE_STATIC	FINE_DYNAMIC
1	17.16	17.2	16.36	11.87
2	33.86	30.68	20.46	7.64
3	45.57	42.94	23.09	8.94
4	55.69	53.38	22.31	9.58
6	70.4	70.31	18.53	9.57
8	91.94	79.42	17.91	9.3

On my virtual machine (Lubuntu)



And following table represents that data in table format

	COARS_STATIC	COARS_DYNAMIC	FINE_STATIC	FINE_DYNAMIC
1	17.16	17.2	16.36	11.87
2	33.86	30.68	20.46	7.64
3	45.57	42.94	23.09	8.94
4	55.69	53.38	22.31	9.58
6	70.4	70.31	18.53	9.57
8	91.94	79.42	17.91	9.3

3. What patterns are you seeing in the speeds?

Answer: Observation: I observed that the overall performance of the Coarse is better than Fine. However, we can observe that in the figure 1 (IPCC with the 16 threads), after a particular thread point (in my graph it is 12), i.e performance at number of threads= 12 is the peak by both the Coars (static and dynamic) scheduling and after that performance is getting slower.

Comparison between dynamic and static: we can observe from the graph that the static scheduling is better than the dynamic in term of performance.

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

Answer: Because Coarse-grained parallelism breaks the task into small number of large task however Fine-grained parallelism breaks task into small task only. And Static scheduling divided the task into N available threads equally whereas the dynamic scheduling divide the task into the some smaller task than the static scheduling and once it is done by one thread then it provides the remaining task to the threads available to do that.

in our scenario, as you can see in the figure that at some point dynamic performs better than static, so at that situation the task assigned to the threads are done quickly and then new task is assigned. However, it did not work all the time.