Multicore Computing Project3

- Problem 1 -

과목명	멀티코어컴퓨팅
제출일	2023.05.22.
학 번	20183901
학 과	소프트웨어학부
이 름	김상민

목 차

1. Environment

Model: MacBook Air(M1, 2020)

Chip: Apple M1

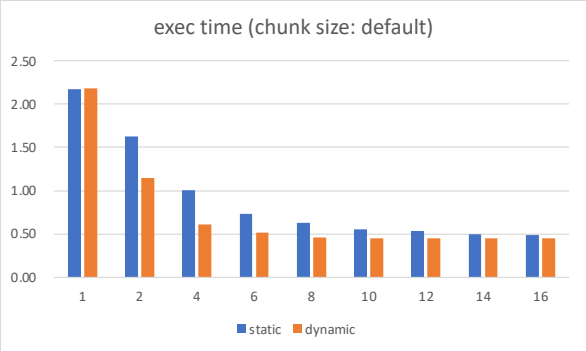
Cores: 4+4 (P: 4 / E: 4)

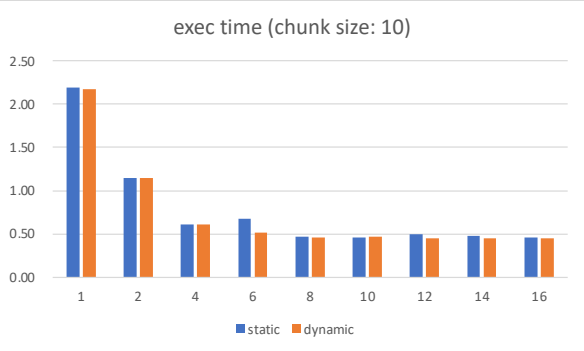
RAM: 16 GB

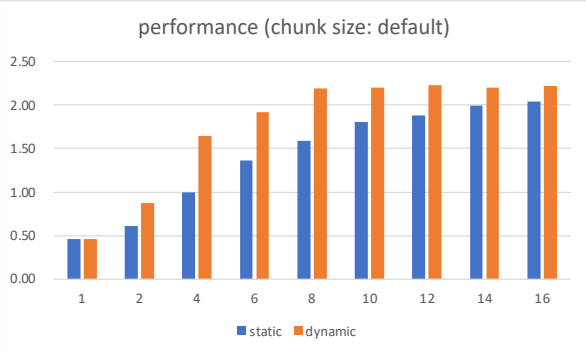
2. Result

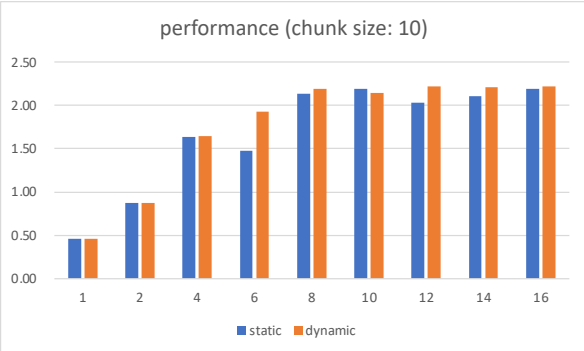
exec time (unit: ms)	chunk size	1	2	4	6	8	10	12	14	16
static	default	2.17	1.63	1.01	0.74	0.63	0.55	0.53	0.50	0.49
dynamic	default	2.18	1.14	0.61	0.52	0.46	0.45	0.45	0.45	0.45
static	10	2.19	1.14	0.61	0.68	0.47	0.46	0.49	0.48	0.46
dynamic	10	2.17	1.14	0.61	0.52	0.46	0.47	0.45	0.45	0.45

performance (1/exec time)	chunk size	1	2	4	6	8	10	12	14	16
static	default	0.46	0.61	0.99	1.36	1.59	1.81	1.88	1.99	2.04
dynamic	default	0.46	0.87	1.64	1.92	2.19	2.20	2.23	2.20	2.22
static	10	0.46	0.87	1.64	1.48	2.14	2.19	2.03	2.10	2.19
dynamic	10	0.46	0.88	1.64	1.93	2.19	2.14	2.22	2.21	2.22









3. Explanation

If we look at the result, overall execution time showed decreasing trend and performance showed the opposite.

When chunk size is default, dynamic method showed better performance than static method, while when chunk size is set 10, both dynamic method and static method showed similar performance.

Also, when we look at static methods, size 10 showed better performance than default. This is because generally default size is larger than 10 (default size = number of loops / number of threads). Using large chunk size will lead to worse load balancing in static methods.