

Multicore Computing Project 3

Hardware and Software Information

Hardware Model	Lenovo ThinkPad T470s
Memory	7.5 GiB
Processor	Intel® Core™ i5-7300U CPU @ 2.60GHz × 4
Graphics	Mesa Intel® HD Graphics 620 (KBL GT2)
Disk Capacity	128.0 GB

OS Name	Fedora Linux 35 (Workstation Edition)
OS Type	64-bit
GNOME Version	41.5
Windowing System	X11
Software Updates	>

Hyperthreading: ON

Core **Count**: 2
Thread **Count**: 4

Problem 1

Tables

Execution Times

exec times in ms	chunk size	1	2	4	6	8	10	12	14	16
static	default	2654	2090	1760	1390	1408	1516	1414	1488	1354
dynamic		2814	2082	1881	1495	1552	1636	1555	1578	1455
exec times in ms	chunk size	1	2	4	6	8	10	12	14	16
static	10	3296	2217	1729	1426	1408	1526	1442	1389	1380
dynamic		2817	2097	1620	1555	1419	1425	1423	1385	1388

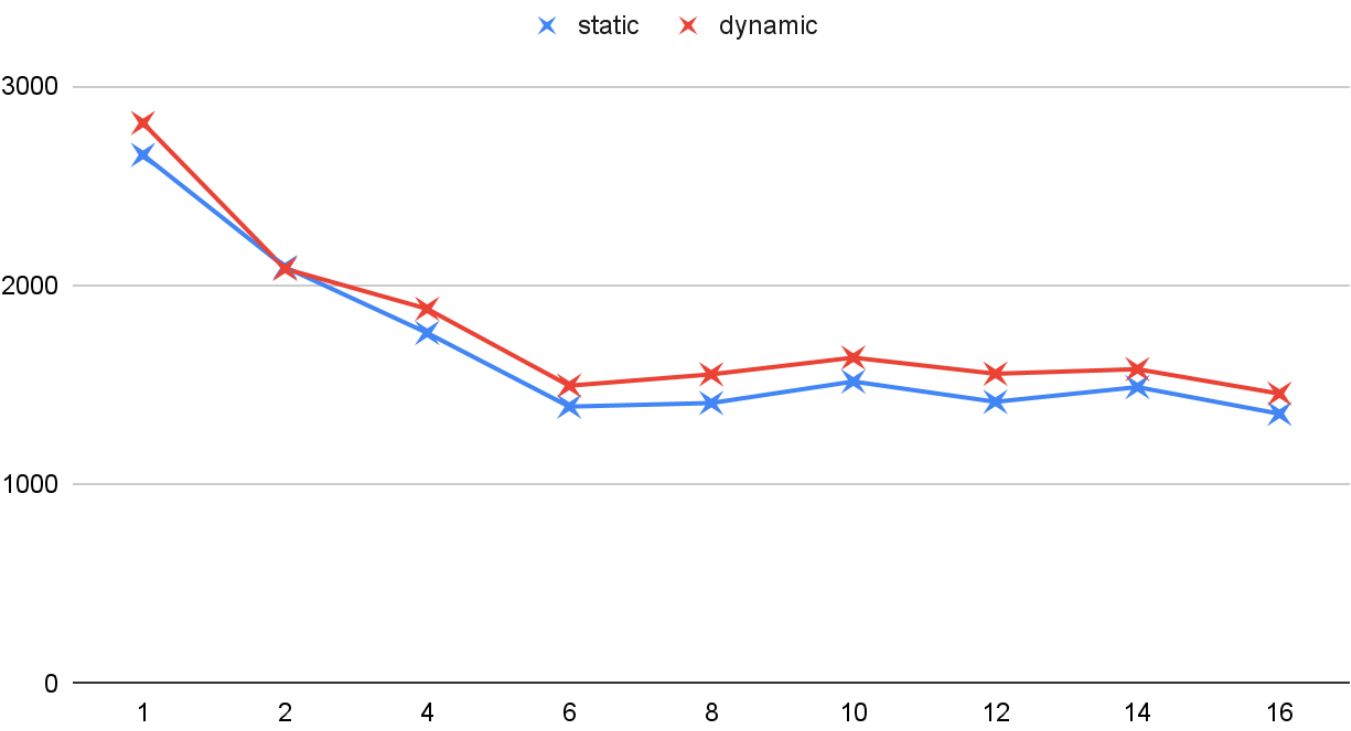
Performance

Perform ance 1/exec time	chunk size	1	2	4	6	8	10	12	14	16
static	default	0,0003767897513	0,0004784688995	0,0005681818182	0,0007194244604	0,0007102272727	0,0006596306069	0,0007072135785	0,0006720430108	0,0007385524372
dynamic		0,000355366027	0,0004803073967	0,0005316321106	0,0006688963211	0,0006443298969	0,0006112469438	0,0006430868167	0,0006337135615	0,0006872852234
Perform ance 1/exec time	chunk size	1	2	4	6	8	10	12	14	16
static	10	0,0003033980583	0,000451059991	0,0005783689994	0,0007012622721	0,0007102272727	0,0006553079948	0,000693481276	0,0007199424046	0,0007246376812
dynamic		0,0003549875754	0,0004768717215	0,0006172839506	0,0006430868167	0,000704721635	0,000701754386	0,0007027406887	0,0007220216606	0,0007204610951

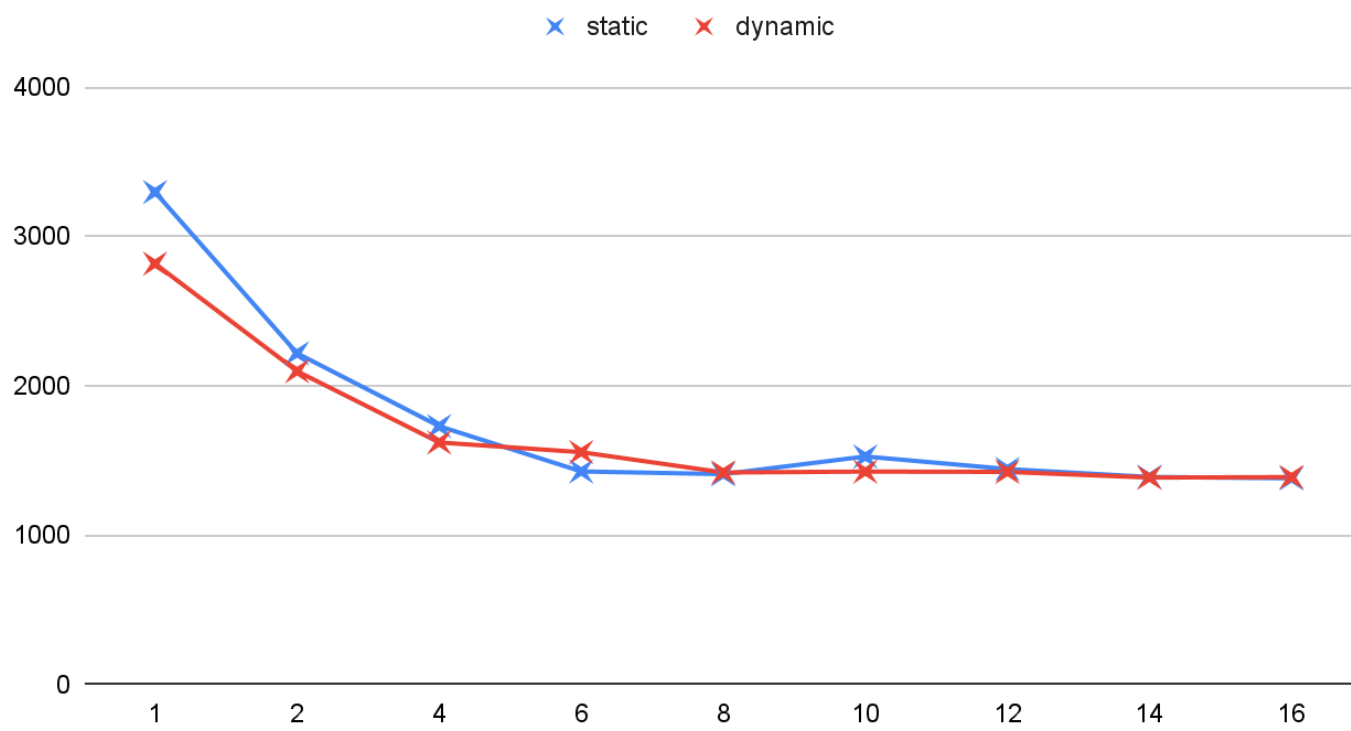
Graphs

Execution Times

Default chunk size

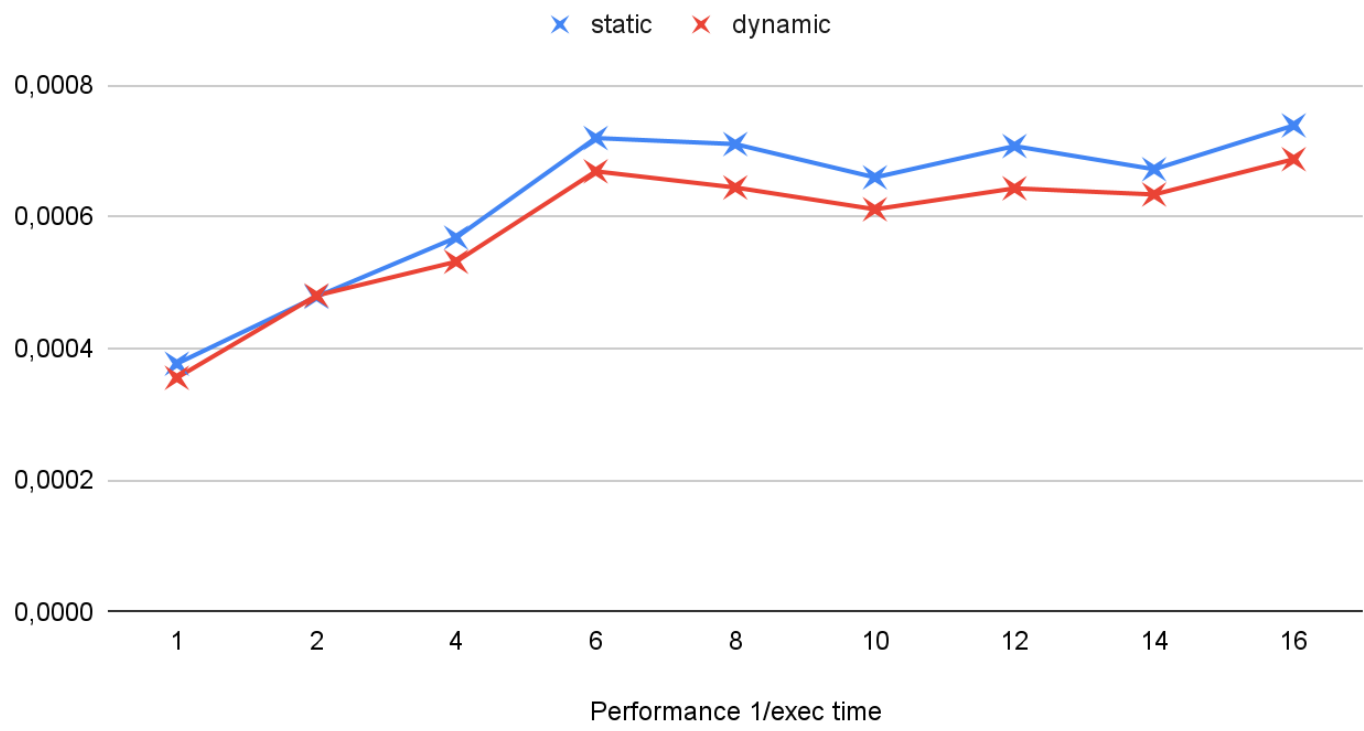


Chunk size of 10

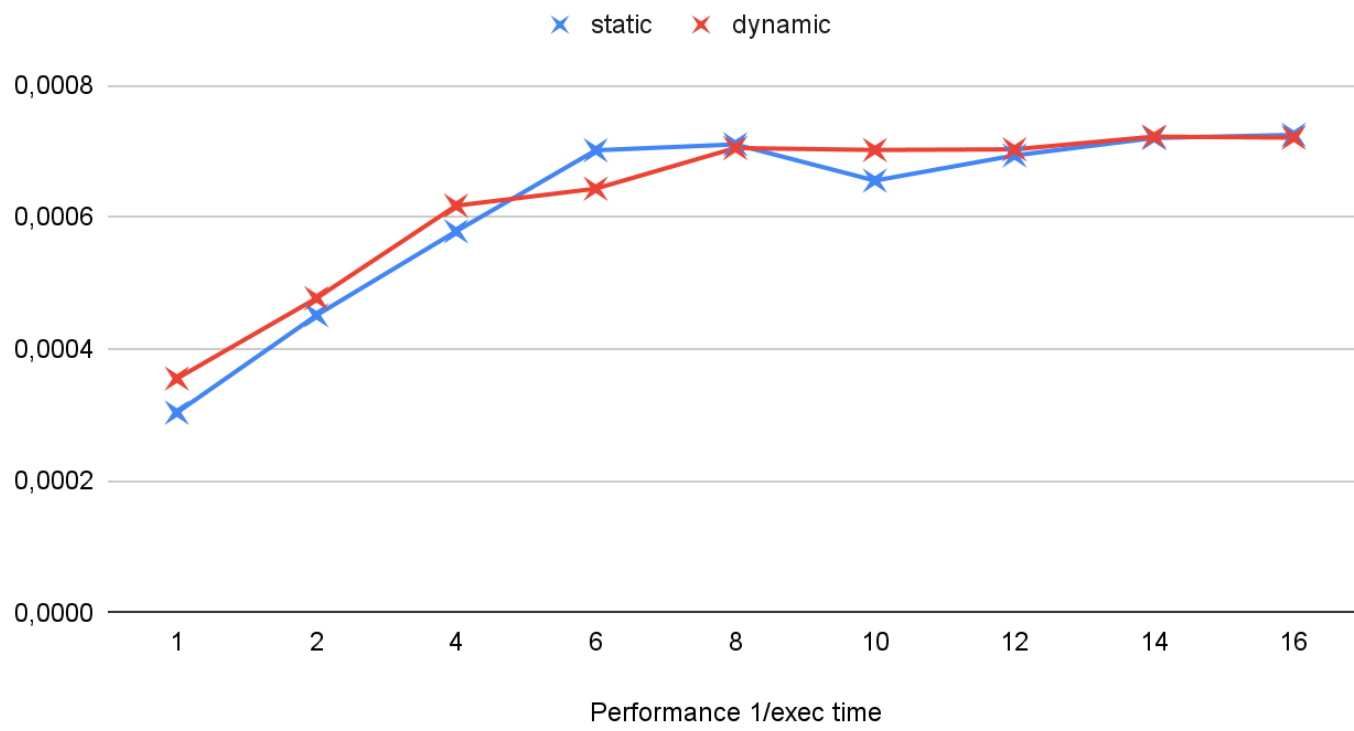


Performance

Default chunk size



Chunk size of 10



Interpretation

My interpretation of these results is that increasing the number of threads and the chunk size, significantly reduces execution times and therefore increases performance. However, it reaches a point of what I would call “peak necessary performance” where execution times and performance do not vary much when adding more threads. As we can see, execution times with 8 threads are rather low and when doubling the number threads results are pretty much the same.

As for comparing the two methods, I think two methods seem to achieve pretty close or identical execution times. So I think the best approach would then come to which one uses less memory.