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 2

Tables

Execution Times

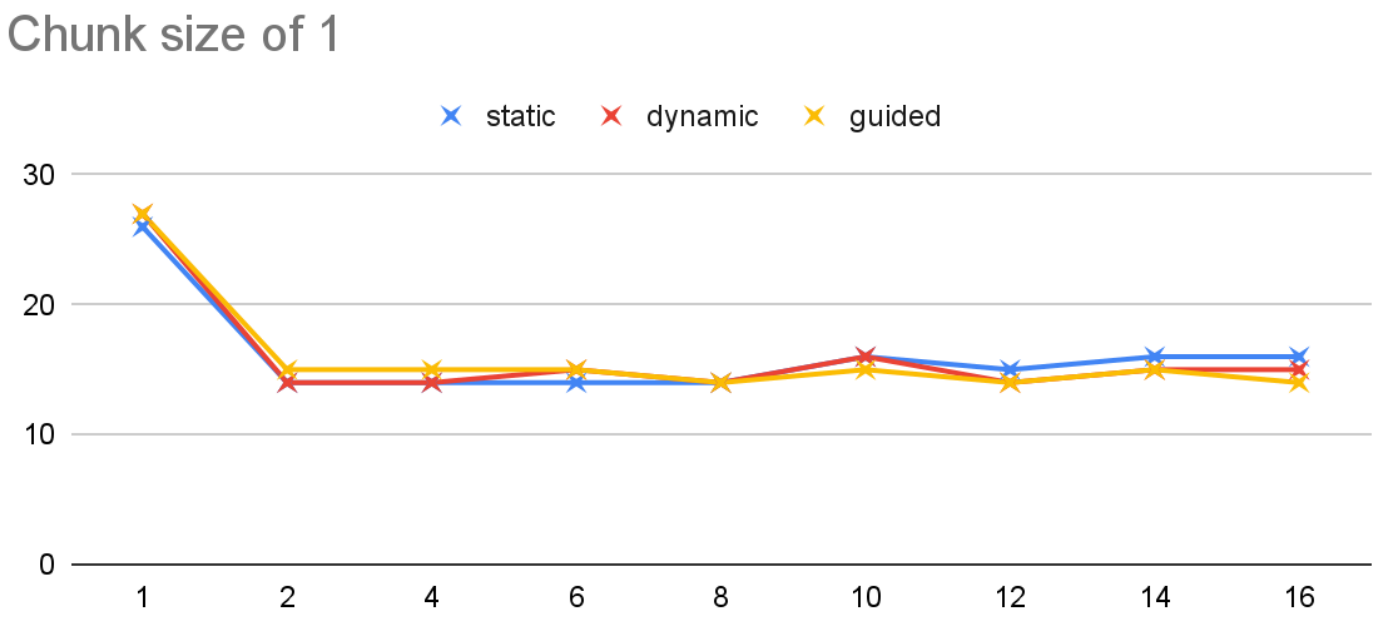
exec times in ms	chunk size	1	2	4	6	8	10	12	14	16
static	1	26	14	14	14	14	16	15	16	16
dynamic		27	14	14	15	14	16	14	15	15
guided		27	15	15	15	14	15	14	15	14
exec times in ms	chunk size	1	2	4	6	8	10	12	14	16
static	5	26	14	14	14	14	14	14	15	14
dynamic		26	14	14	19	15	15	15	15	14
guided		26	14	14	14	14	15	14	14	14
exec times in ms	chunk size	1	2	4	6	8	10	12	14	16
static	10	26	14	15	14	15	14	15	15	14
dynamic		26	25	15	15	14	15	14	15	14
guided		26	14	15	14	14	15	14	15	15
exec times in ms	chunk size	1	2	4	6	8	10	12	14	16
static	100	26	14	14	17	15	15	14	15	14
dynamic		26	14	14	15	15	18	18	17	15
guided		35	15	15	15	15	16	15	16	17

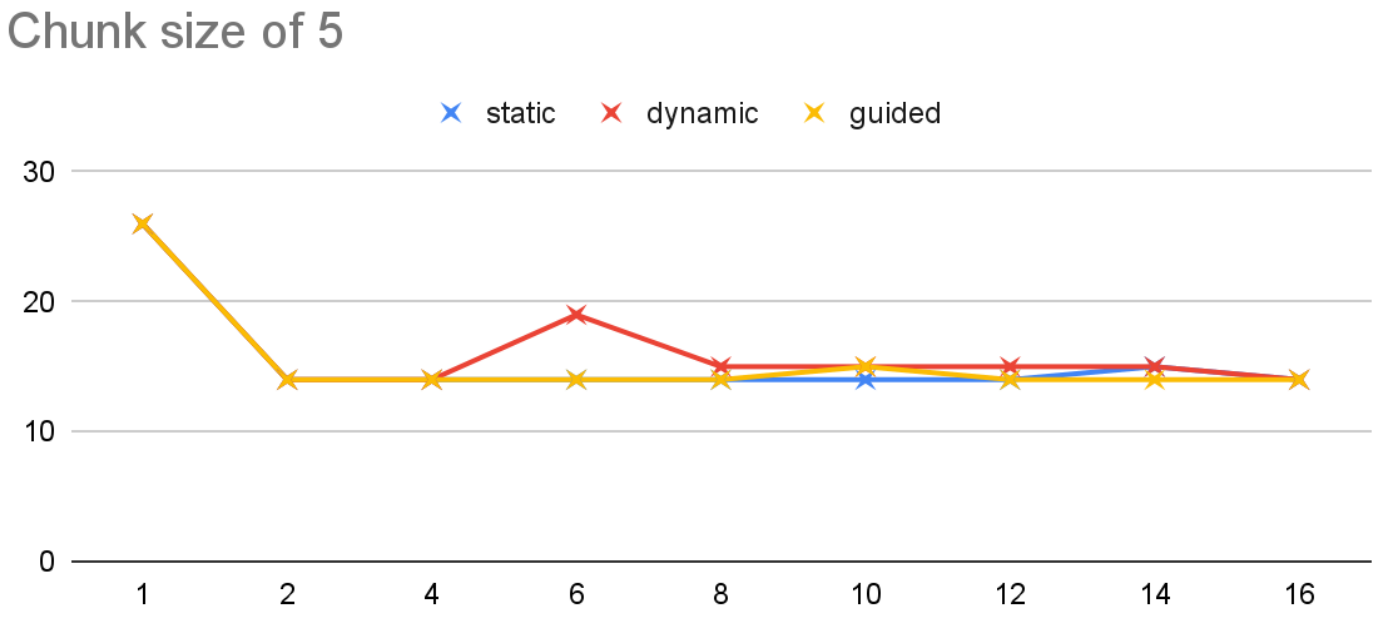
Performance

Performance (1/exec time)	chunk size	1	2	4	6	8	10	12	14	16
static	1	0,03846153846	0,07142857143	0,07142857143	0,07142857143	0,07142857143	0,0625	0,06666666667	0,0625	0,0625
dynamic		0,03703703704	0,07142857143	0,07142857143	0,06666666667	0,07142857143	0,0625	0,07142857143	0,06666666667	0,06666666667
guided		0,03703703704	0,06666666667	0,06666666667	0,06666666667	0,07142857143	0,06666666667	0,07142857143	0,06666666667	0,07142857143
Performance (1/exec time)	chunk size	1	2	4	6	8	10	12	14	16
static	5	0,03846153846	0,07142857143	0,07142857143	0,07142857143	0,07142857143	0,07142857143	0,07142857143	0,06666666667	0,07142857143
dynamic		0,03846153846	0,07142857143	0,07142857143	0,05263157895	0,06666666667	0,06666666667	0,06666666667	0,06666666667	0,07142857143
guided		0,03846153846	0,07142857143	0,07142857143	0,07142857143	0,07142857143	0,06666666667	0,07142857143	0,07142857143	0,07142857143
Performance (1/exec time)	chunk size	1	2	4	6	8	10	12	14	16
static	10	0,03846153846	0,07142857143	0,06666666667	0,07142857143	0,06666666667	0,07142857143	0,06666666667	0,06666666667	0,07142857143
dynamic		0,03846153846	0,04	0,06666666667	0,06666666667	0,07142857143	0,06666666667	0,07142857143	0,06666666667	0,07142857143
guided		0,03846153846	0,07142857143	0,06666666667	0,07142857143	0,07142857143	0,06666666667	0,07142857143	0,06666666667	0,06666666667
Performance (1/exec time)	chunk size	1	2	4	6	8	10	12	14	16
static	100	0,03846153846	0,07142857143	0,07142857143	0,05882352941	0,06666666667	0,06666666667	0,07142857143	0,06666666667	0,07142857143
dynamic		0,03846153846	0,07142857143	0,07142857143	0,06666666667	0,06666666667	0,05555555556	0,05555555556	0,05882352941	0,06666666667
guided		0,02857142857	0,06666666667	0,06666666667	0,06666666667	0,06666666667	0,0625	0,06666666667	0,0625	0,05882352941

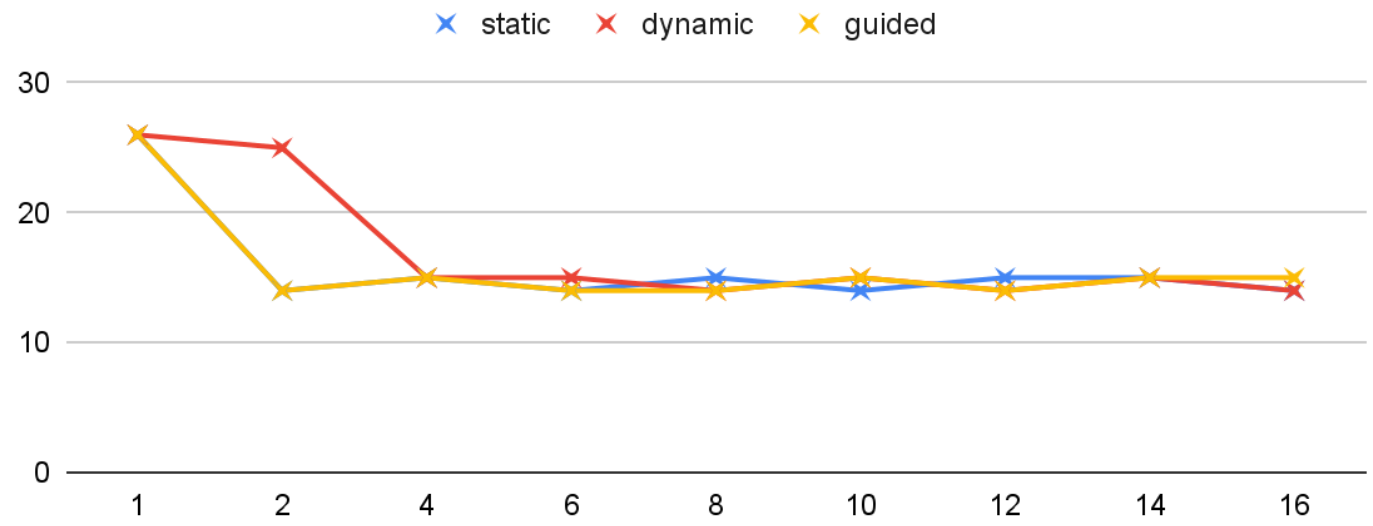
Graphs

Execution Times

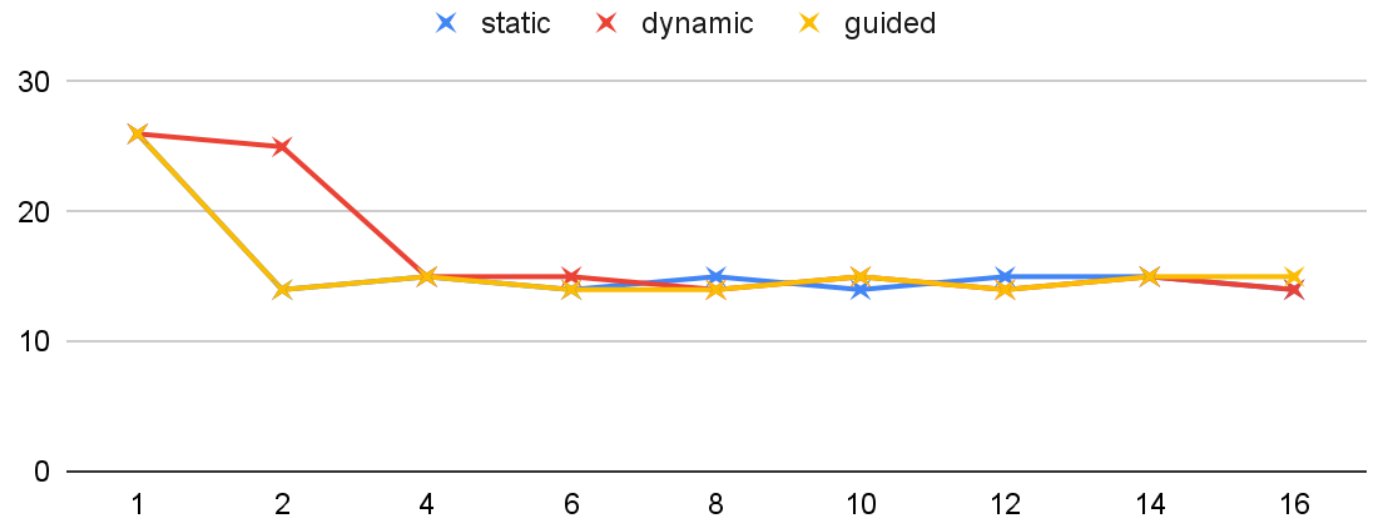




Chunk size of 10

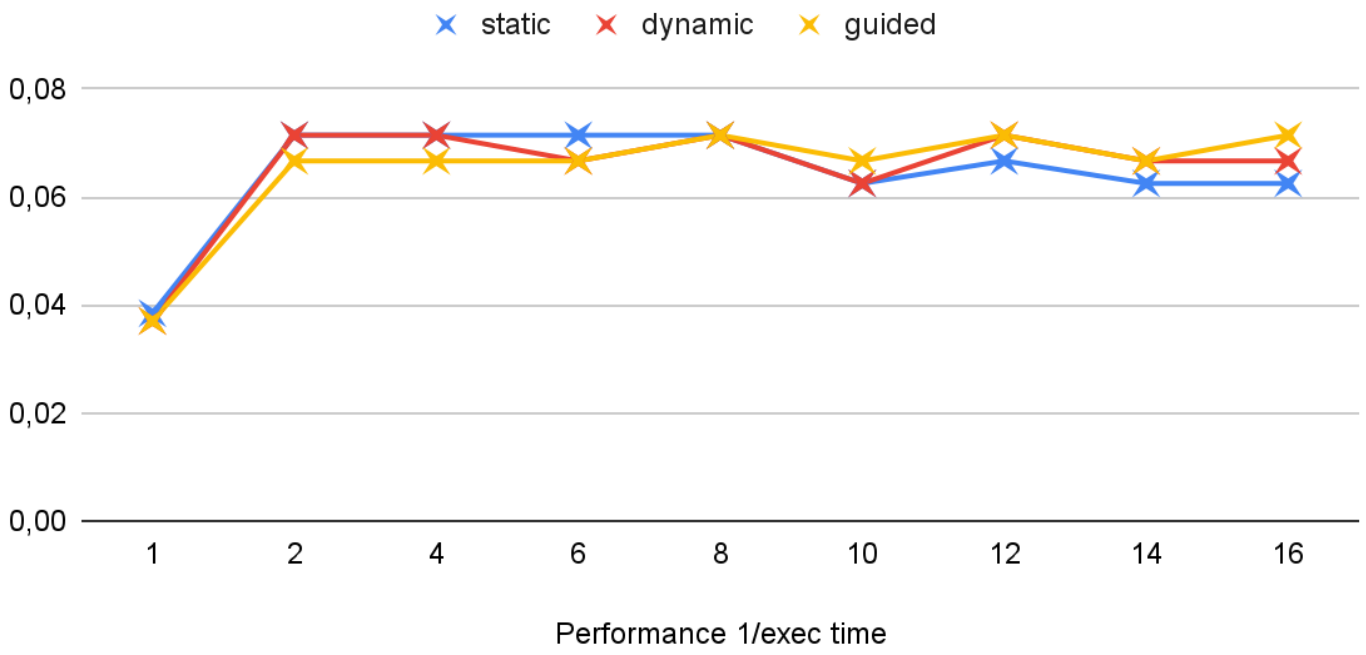


Chunk size of 100

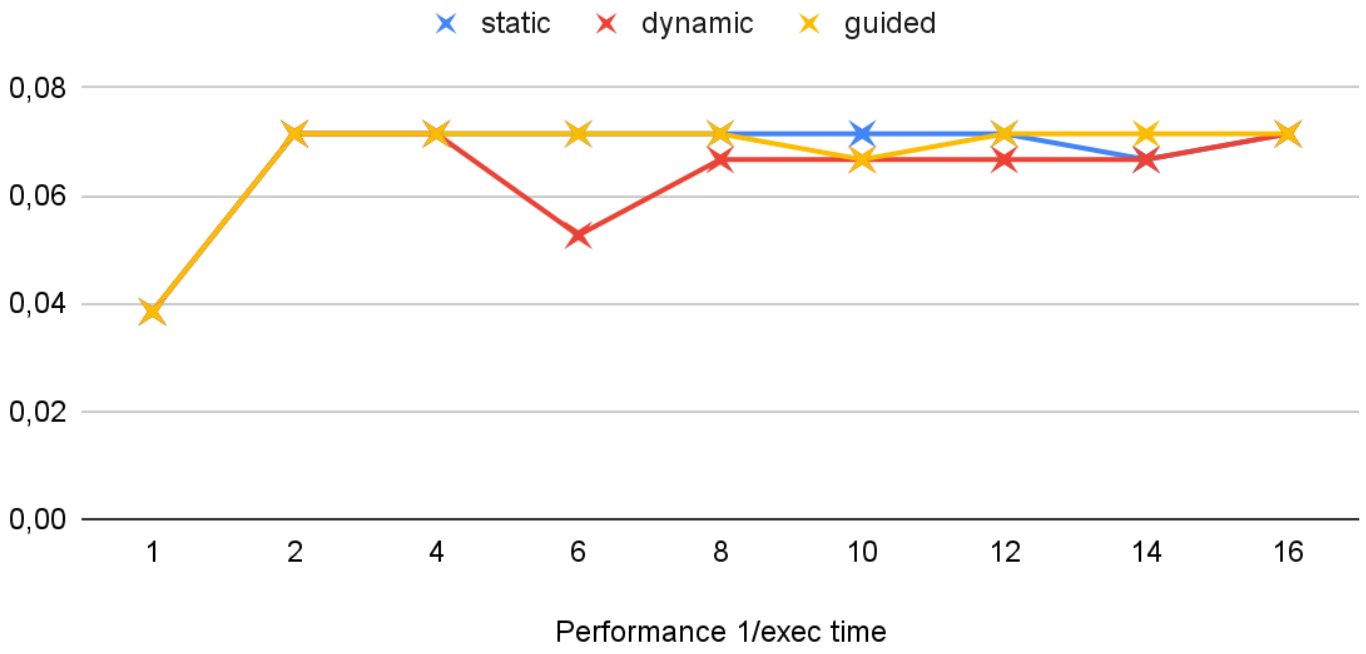


Performance

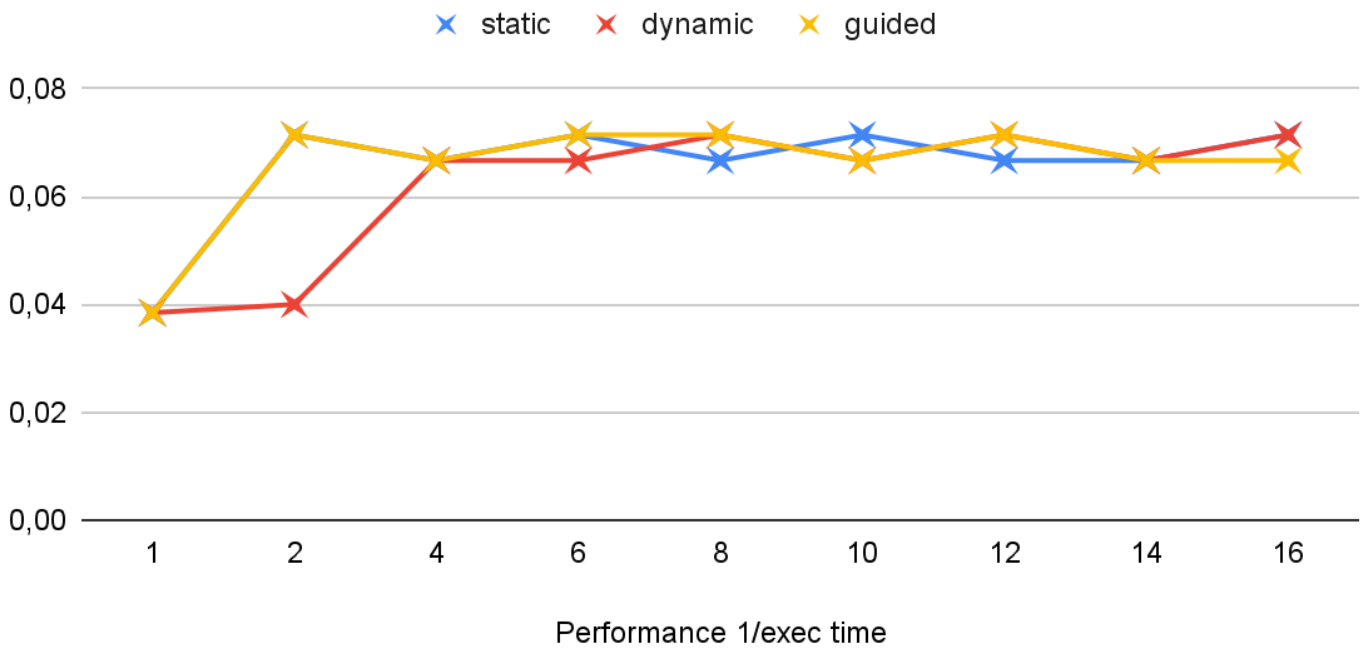
Chunk size of 1



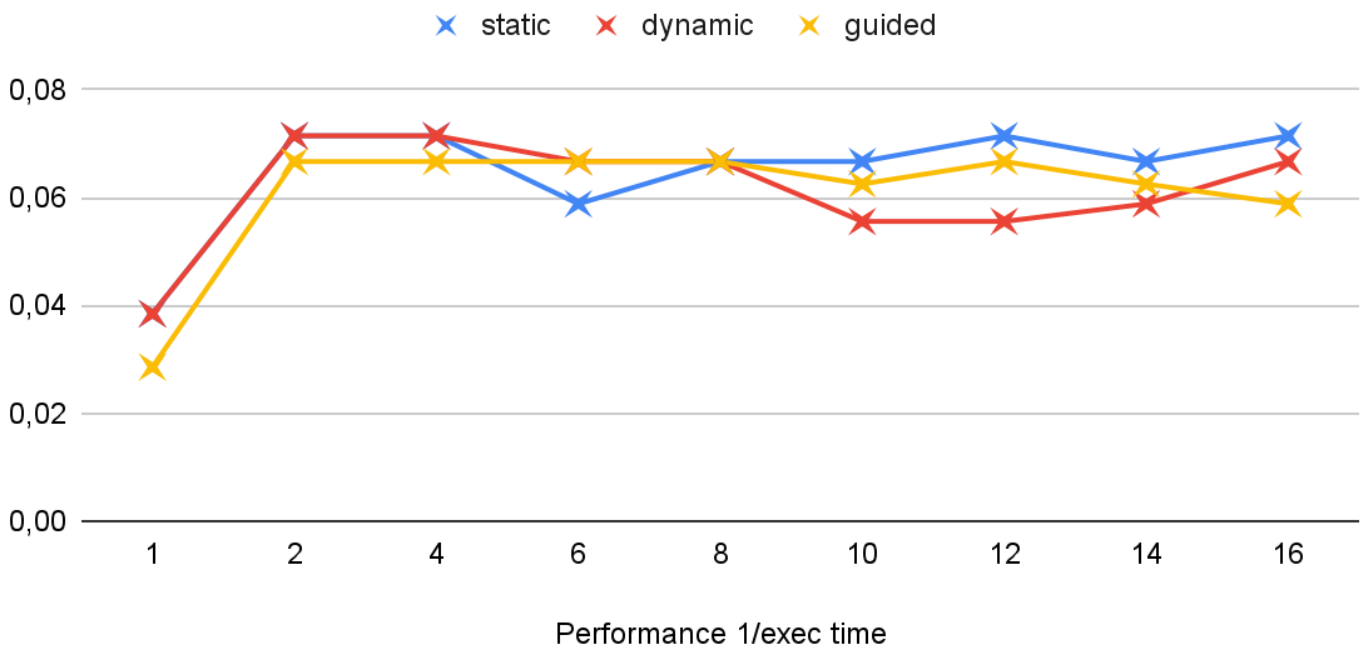
Chunk size of 5



Chunk size of 10



Chunk size of 100



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