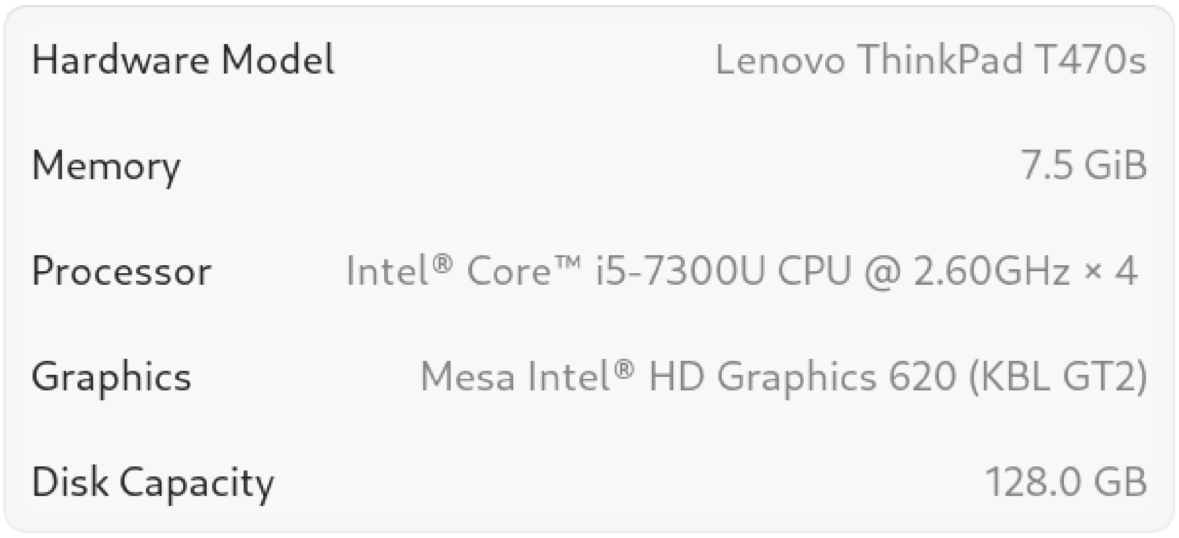
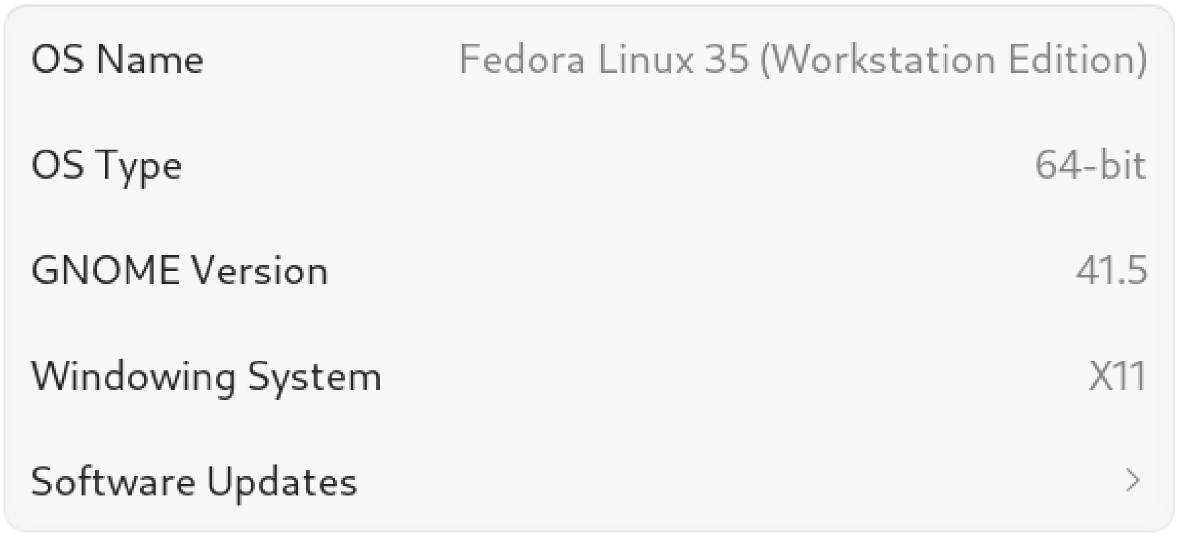
Multicore Computing Project 1

Hardware and Software Information

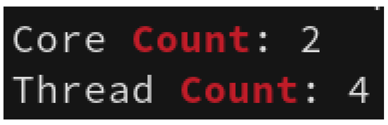


Hardware Model	Lenovo ThinkPad T470s
Memory	7.5 GiB
Processor	Intel® Core™ i5-7300U CPU @ 2.60GHz × 4
Graphics	Mesa Intel® HD Graphics 620 (KBL GT2)
Disk Capacity	128.0 GB



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

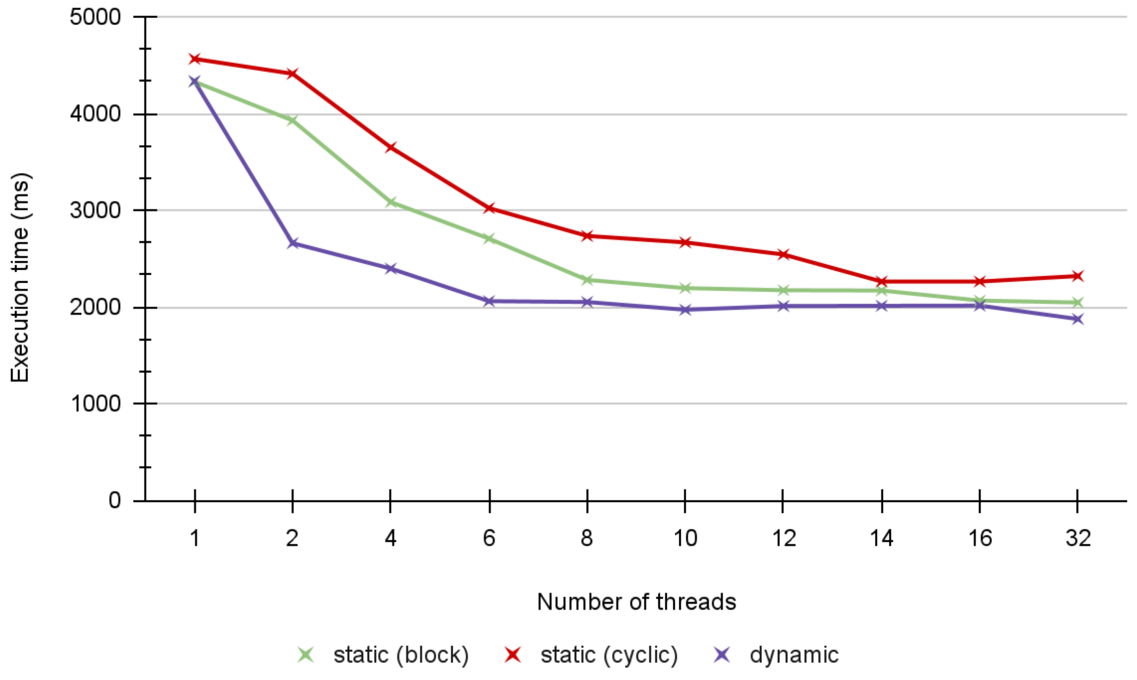
exec times in ms	1	2	4	6	8	10	12	14	16	32
static (block)	4330	3929	3085	2707	2280	2195	2173	2170	2066	2046
static (cyclic)	4566	4412	3650	3023	2735	2668	2543	2263	2263	2320
dynamic	4336	2660	2397	2060	2051	1971	2010	2012	2014	1875

Performance

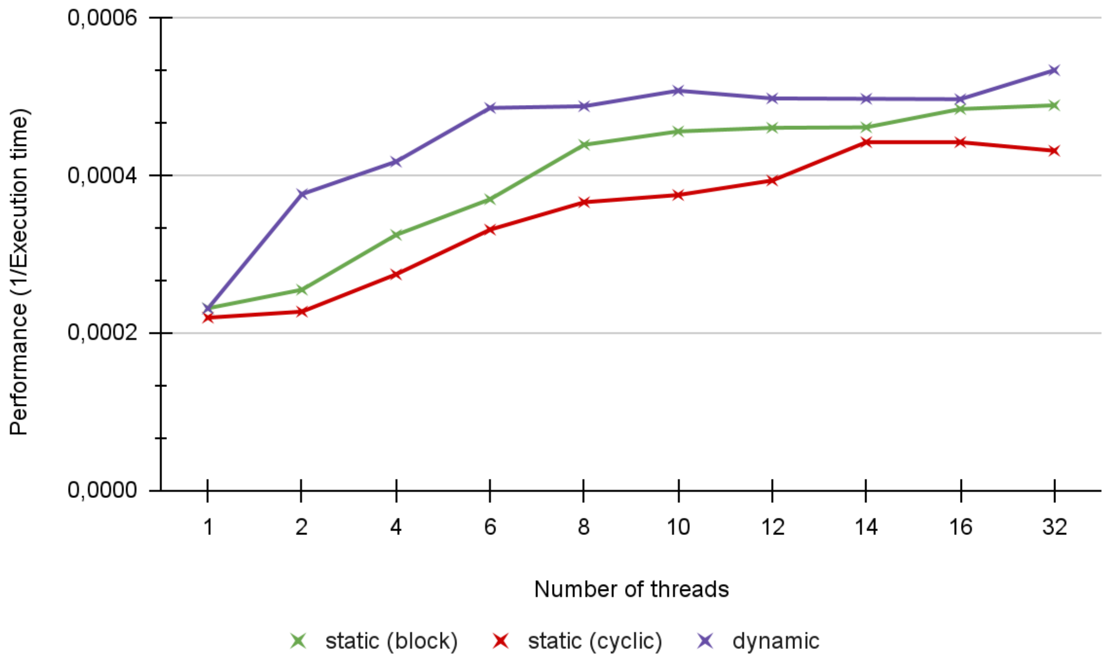
Performance 1/exec time	1	2	4	6	8	10	12	14	16	32
static (block)	0,00023094688 22	0,00025451768 9	0,00032414910 86	0,00036941263 39	0,00043859649 12	0,00045558086 56	0,00046019328 12	0,00046082949 31	0,00048402710 55	0,00048875855 33
static (cyclic)	0,00021901007 45	0,00022665457 84	0,00027397260 27	0,00033079722 13	0,00036563071 3	0,00037481259 37	0,00039323633 5	0,00044189129 47	0,00044189129 47	0,00043103448 28
dynamic	0,00023062730 63	0,00037593984 96	0,00041718815 19	0,00048543689 32	0,00048756704 05	0,00050735667 17	0,00049751243 78	0,00049701789 26	0,00049652432 97	0,00053333333 33

Graphs

Execution Times



Performance



Interpretation

My interpretation of these results is that increasing the number of threads, 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 16 threads are rather low and when doubling the number threads results are pretty much the same.
As for comparing the three methods, I think dynamic load balancing is definitely the best approach to a problem like this one, when the number of threads is low. However when increasing the number of threads the three methods seem to achieve pretty close or identical, sometimes even better execution times. So I think the best approach would then come to which one uses less memory.

Problem 2

Tables

Execution Times

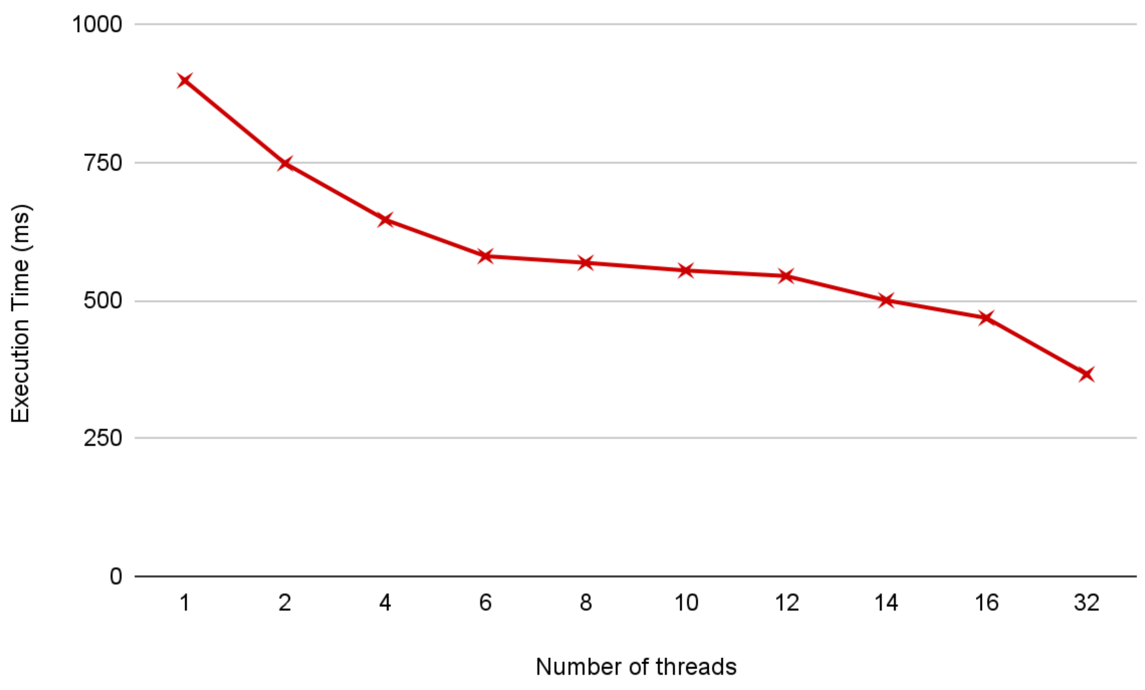
	1	2	4	6	8	10	12	14	16	32
exec times in ms	449	374	323	290	284	277	272	250	234	183

Performance

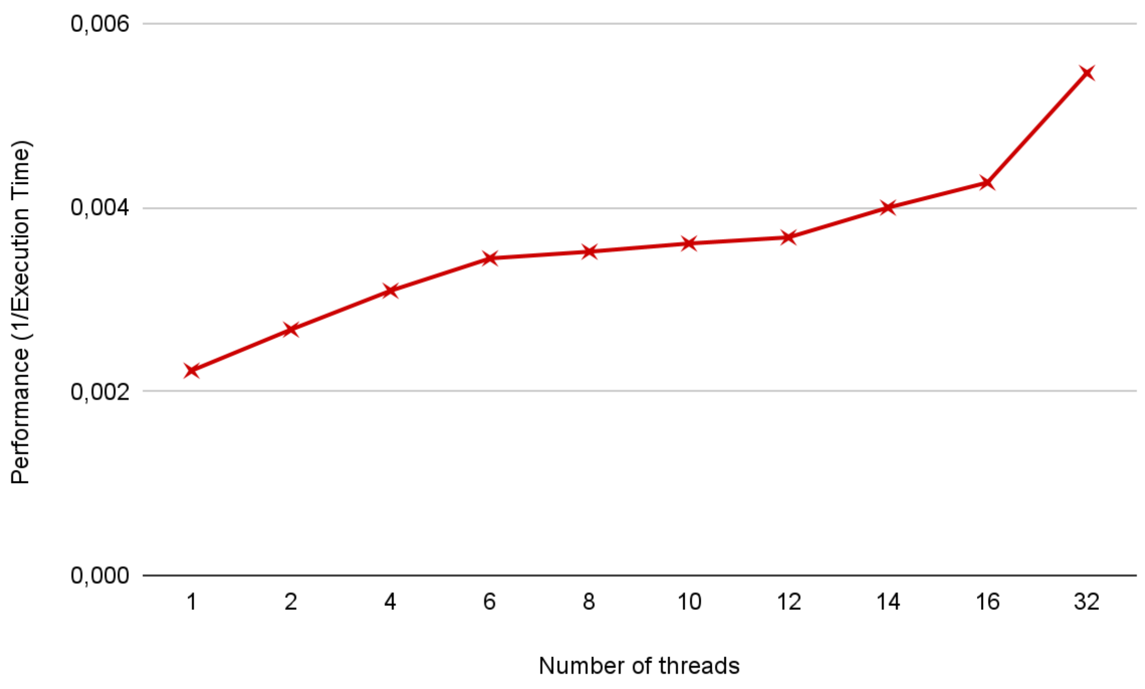
	1	2	4	6	8	10	12	14	16	32
Performance 1/exec time	0,002227171492	0,002673796791	0,003095975232	0,003448275862	0,003521126761	0,003610108303	0,003676470588	0,004	0,004273504274	0,005464480874

Graphs

Execution Times



Performance


Performance (1/Execution Time)
Number of threads