Exercise 2

1D

T=N*500

seq	2perhost 2	8perhost 8
2	1	1
9	6	3
41	24	9
65	37	13

N	speedup 2per	speedup 8perh	efficiency 2perh	efficiency 8perhost 8
1000	2	2	0.5	0.25
2000	1.5	3	0.375	0.375
4000	1.708333333	4.5555555556	0.42708333333	0.569444444444444
5000	1.756756757	5	0.43918918919	0.625

2D

T=N*100

N	seq	4perhost 4	speedup	efficiency
100	5	1	5	1.25
200	45	12	3.75	0.9375
250	90	24	3.75	0.9375
300	154	95	1.62105263158	0.405263157894737

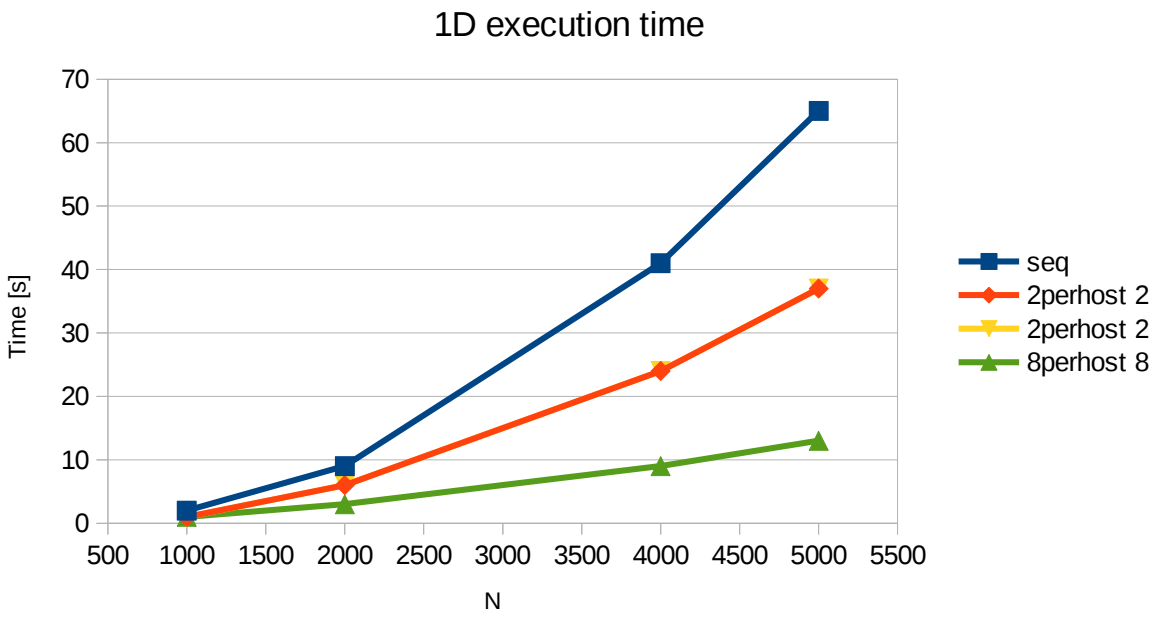
3D

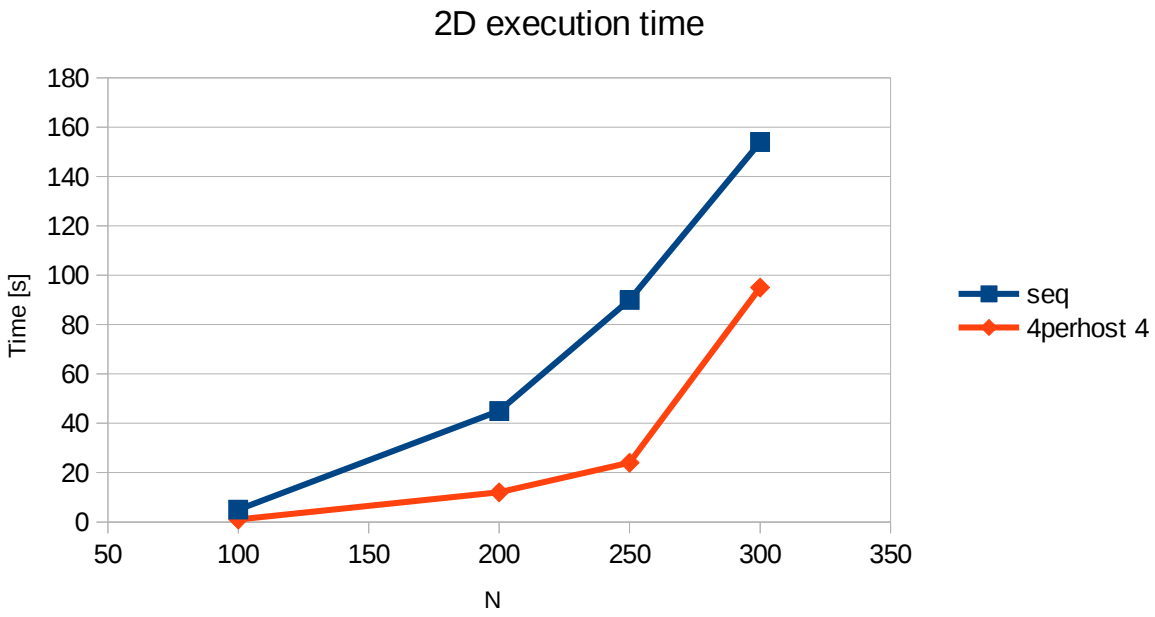
T=N*20

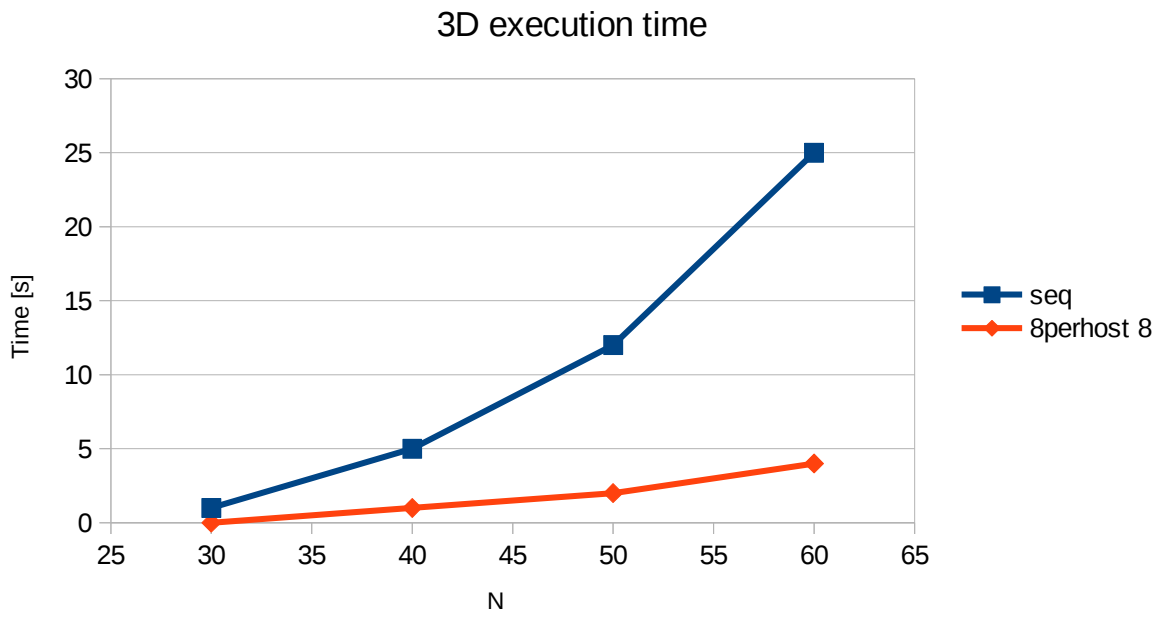
N	seq	8perhost 8	speedup	efficiency
30	1	0	#DIV/0!	#DIV/0!
40	5	1	5	0.625
50	12	2	6	0.75
60	25	4	6.25	0.78125

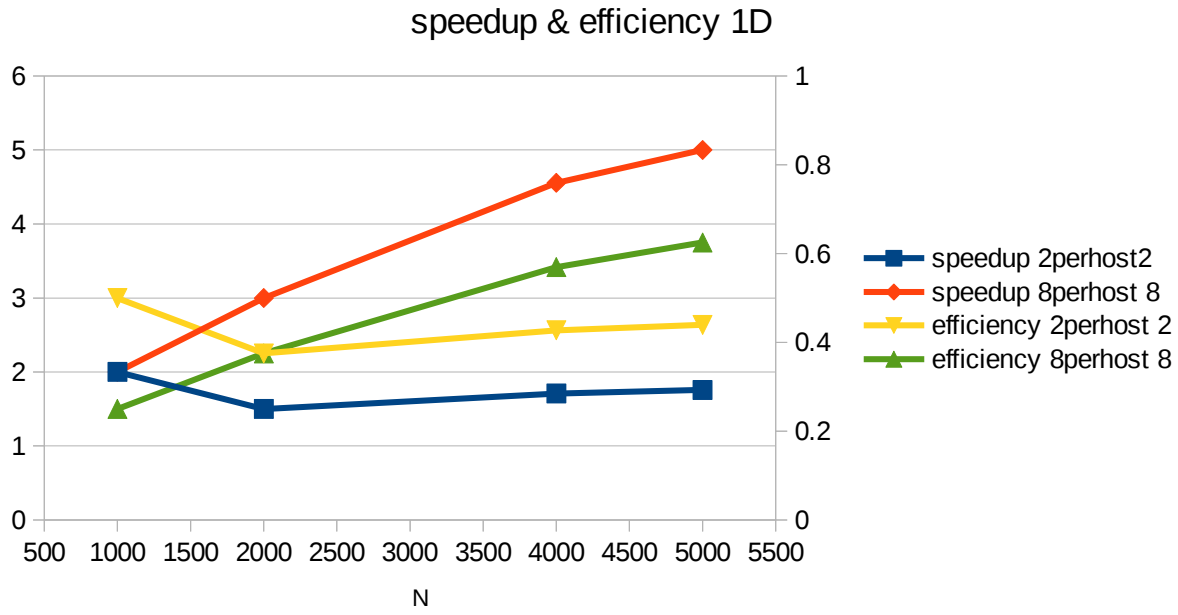
Notes:
speedup here means absolute speedup → reference is fastest sequential version

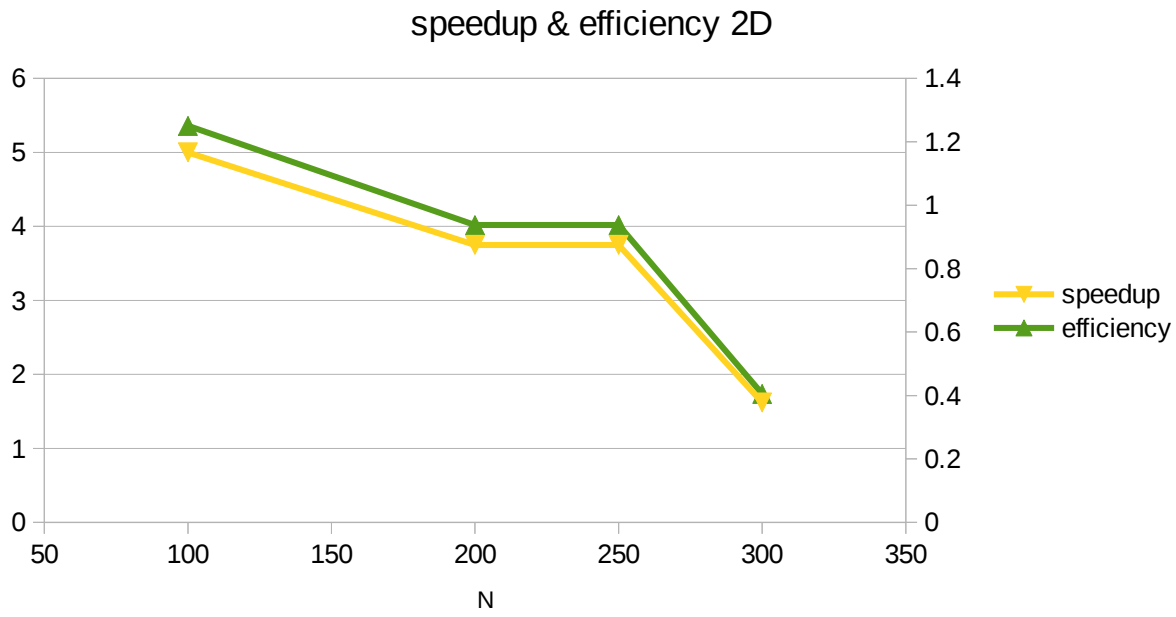
We used strong and weak scalability. Per row we compared execution time on a fixed size (strong) problem. In each column we sized up the problem (weak) and compared results.

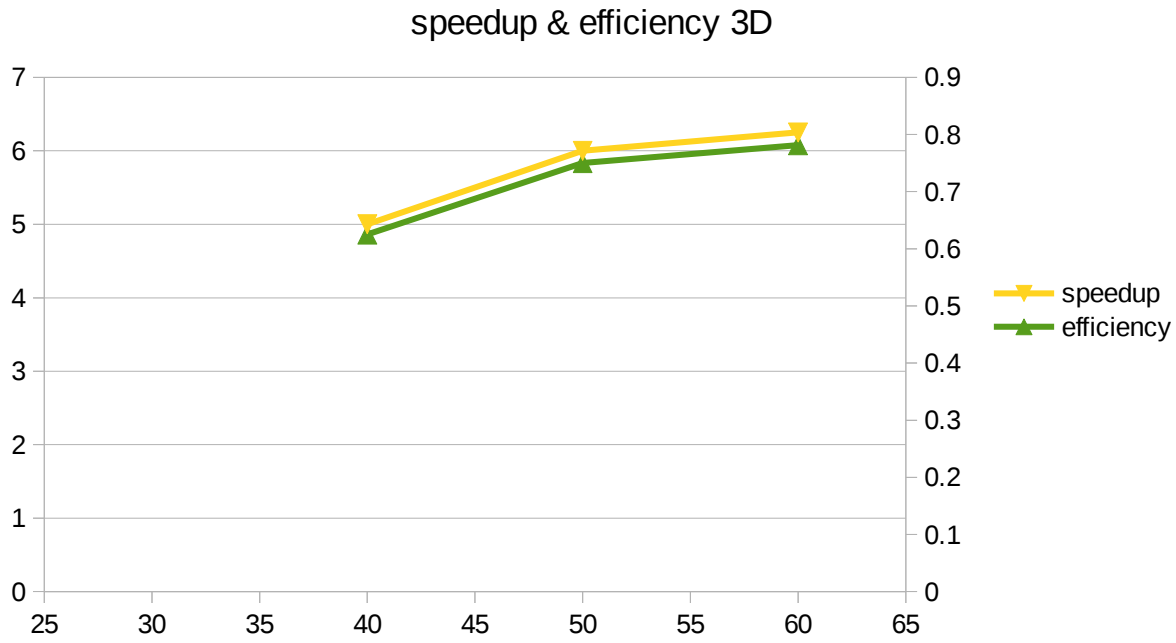

1D execution time
Time [s]
70
60
50
40
30
20
10
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
N
seq
2perhost 2
2perhost 2
8perhost 8


2D execution time
Time [s]
180
160
140
120
100
80
60
40
20
0
50
100
150
200
250
300
350
N
seq
4perhost 4


3D execution time
Time [s]
30
25
20
15
10
5
0
25
30
35
40
45
50
55
60
65
N
seq
8perhost 8


speedup & efficiency 1D
6
5
4
3
2
1
0
1
0.8
0.6
0.4
0.2
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
N
speedup 2perhost2
speedup 8perhost 8
efficiency 2perhost 2
efficiency 8perhost 8


speedup & efficiency 2D
6
5
4
3
2
1
0
1.4
1.2
1
0.8
0.6
0.4
0.2
0
50
100
150
200
250
300
350
N
speedup
efficiency


speedup & efficiency 3D
7
6
5
4
3
2
1
0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
25
30
35
40
45
50
55
60
65
speedup
efficiency