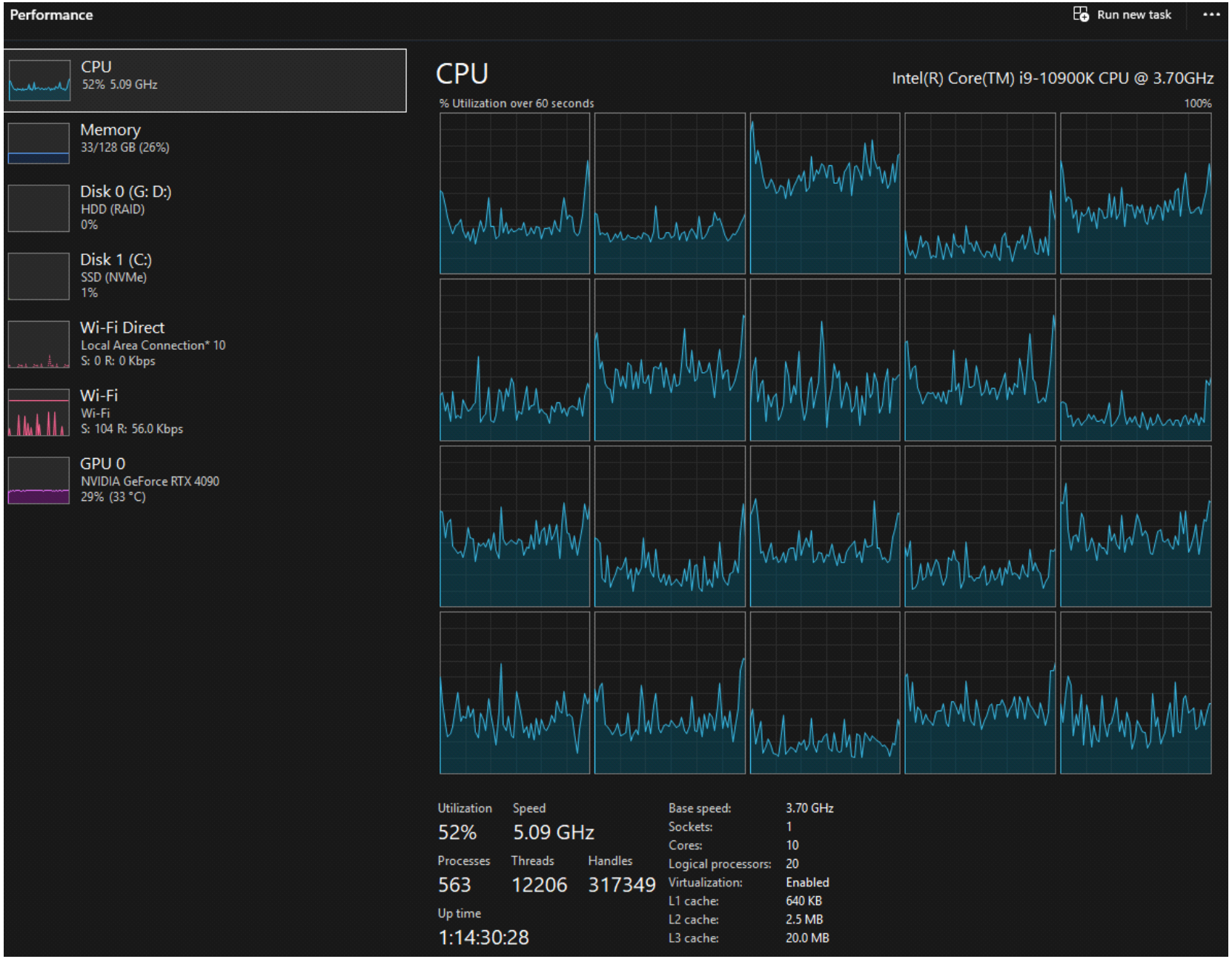
PC stats

Saturday, 8 March 2025 10:13



Darts per thread performance

Saturday, 8 March 2025 10:13

final results after 10 iterations:
RT_THREADS: 20
DARTS_PER_THREAD_RT: 20
AVG Pi: 3.1426752000000002
AVG Time: 0.0067997299999999995 seconds

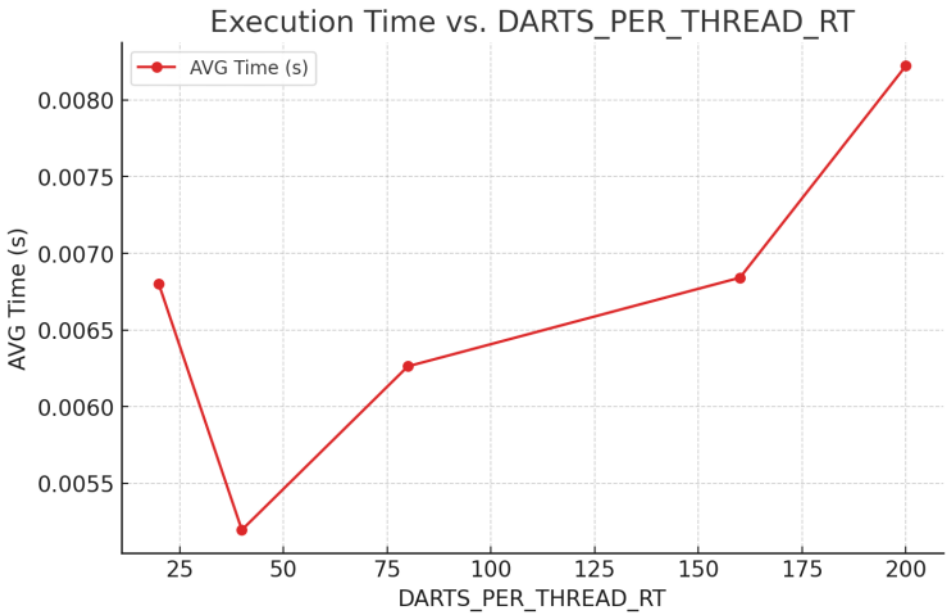
RT_THREADS: 20
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1425912
AVG Time: 0.005196179800000001 seconds

RT_THREADS: 20
DARTS_PER_THREAD_RT: 80
AVG Pi: 3.1429712
AVG Time: 0.0062631697 seconds

RT_THREADS: 20
DARTS_PER_THREAD_RT: 160
AVG Pi: 3.1379656000000002
AVG Time: 0.006841280399999999 seconds

RT_THREADS: 20
DARTS_PER_THREAD_RT: 200
AVG Pi: 3.1413744
AVG Time: 0.0082236699 seconds

DARTS_PER_THREAD_RT	AVG_Pi	AVG_Time (s)
20	3.1426752	0.00679973
40	3.1425912	0.00519618
80	3.1429712	0.00626317
160	3.1379656	0.00684128
200	3.1413744	0.00822367



I tested different numbers of darts per thread to figure out the sweet spot for the best performance
here's what I found:

- **Around 40 darts per thread** - we get low execution time and the best efficiency
- **Above 80 darts per thread** - execution time starts to increase giving us diminishing returns
- **At 200 darts per thread, things slow down noticeably** - likely because there's too much work assigned to each thread and it reduces the parallel efficiency

Thread Scaling with JVM warmup (10 iterations)

Saturday, 8 March 2025 10:13

final results after 10 iterations:
RT_THREADS: 1
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1419084
AVG Time: 0.011772790100000001 seconds

RT_THREADS: 2
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1415512
AVG Time: 0.0078122301 seconds

RT_THREADS: 4
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.141828
AVG Time: 0.005230339700000001 seconds

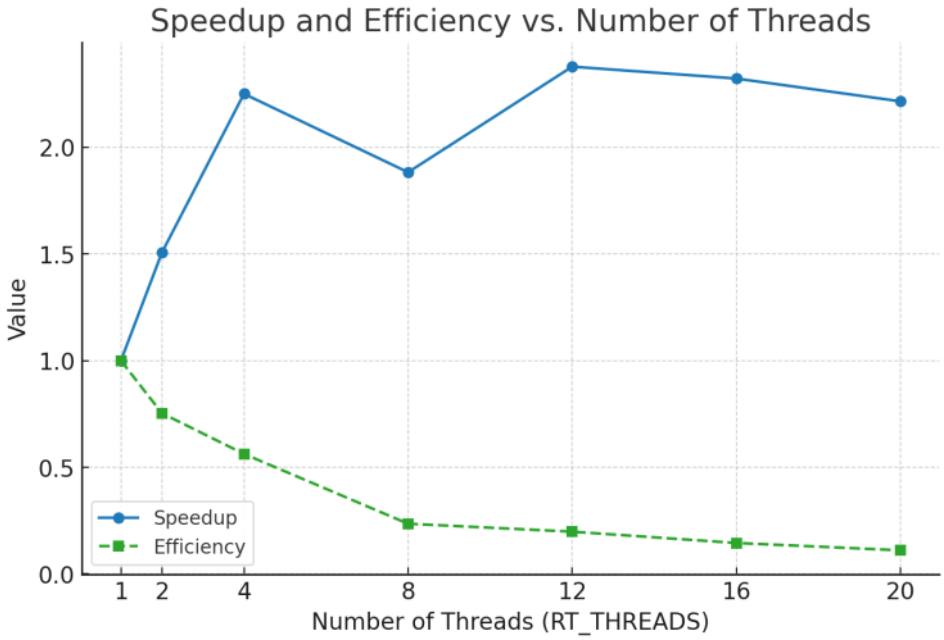
RT_THREADS: 8
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1418972
AVG Time: 0.006251760099999999 seconds

RT_THREADS: 12
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1416032
AVG Time: 0.0049501899000000005 second

RT_THREADS: 16
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.140836
AVG Time: 0.0050687598 seconds

RT_THREADS: 20
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1415948
AVG Time: 0.0053132202 seconds

RT_THREADS	DARTS_PER_THREAD_RT	AVG_Pi	AVG_Time (s)	Speedup	Efficiency
1	40	3.141908	0.011773	1	1
2	40	3.141551	0.007812	1.506969	0.753485
4	40	3.141828	0.00523	2.250865	0.562716
8	40	3.141897	0.006252	1.883116	0.23539
12	40	3.141603	0.00495	2.37825	0.198188
16	40	3.140836	0.005069	2.322617	0.145164
20	40	3.141595	0.005313	2.215754	0.110788



running 10 iterations...
iteration 1 - Pi approx: 3.142248, time: 0.034664499 seconds
iteration 2 - Pi approx: 3.143008, time: 0.0022894 seconds
running 10 iterations...
iteration 1 - Pi approx: 3.142492, time: 0.0308462 seconds
iteration 2 - Pi approx: 3.141696, time: 0.001758799 seconds
running 10 iterations...
iteration 1 - Pi approx: 3.136556, time: 0.0384198 seconds
iteration 2 - Pi approx: 3.142016, time: 0.003612601 seconds
running 10 iterations...
iteration 1 - Pi approx: 3.140988, time: 0.025395301 seconds
iteration 2 - Pi approx: 3.14308, time: 0.0054349 seconds

In the initial implementation of the code, I've noticed that the first iteration always takes 10-20 times longer than all the subsequent ones. I think this is due to the JVM warmup, the thread initializations, the class loading etc.
So I will leave this as is, and make some changes to the code to increase iterations to 11 and exclude the first one from the average calculations, should lead to better more consistent results

Thread Scaling excluding JVM warmup (11 iterations)

Saturday, 8 March 2025 10:13

final results after 11 iterations:
RT_THREADS: 1
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1411688
AVG Time: 0.0095647802 seconds

RT_THREADS: 2
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1410816
AVG Time: 0.0049363799 seconds

RT_THREADS: 4
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1410268
AVG Time: 0.0028015302 seconds

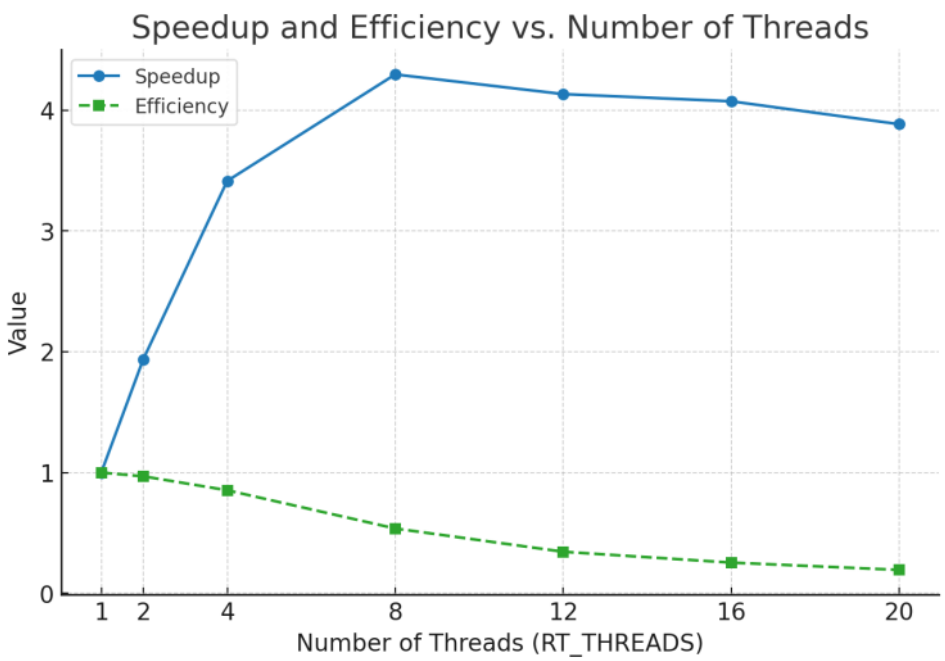
RT_THREADS: 8
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.14229
AVG Time: 0.0022266601 seconds

RT_THREADS: 12
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1416616
AVG Time: 0.0023142799 seconds

RT_THREADS: 16
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1402152
AVG Time: 0.0023481401 seconds

RT_THREADS: 20
DARTS_PER_THREAD_RT: 40
AVG Pi: 3.1410656
AVG Time: 0.0024626496 seconds

RT_THREADS	DARTS_PER_THREAD_RT	AVG_Pi	AVG_Time (s)	Speedup	Efficiency
1	40	3.141169	0.009565	1	1
2	40	3.141082	0.004936	1.93761	0.968805
4	40	3.141027	0.002802	3.414127	0.853532
8	40	3.14229	0.002227	4.295573	0.536947
12	40	3.141662	0.002314	4.13294	0.344412
16	40	3.140215	0.002348	4.073343	0.254584
20	40	3.141066	0.002463	3.883938	0.194197



Takeaways from the data

- Best speedup is at 8 threads, after that it starts declining - meaning diminishing returns in parallelization most likely due to thread management and sync
- Efficiency drops with each added thread - again because of the communication overhead - so more threads does not mean improved performance
- Sweet spot is at 8 threads we get 53% efficiency, a good balance between overhead and parallelism
- Beyond 12 threads efficiency drops under 30% - this means we spend more time on managing threads than on actually computing
- We can see how overhead and communication affects the program at 16 and 20 threads