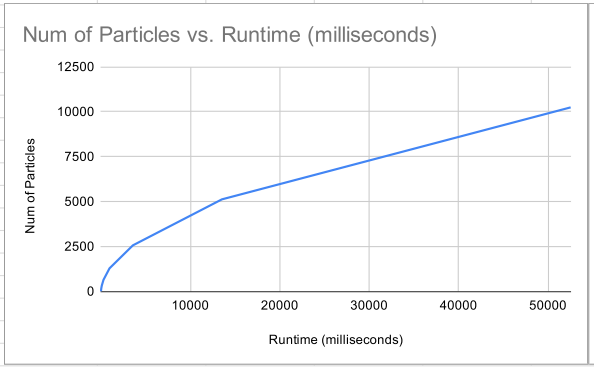
Runtime (millisec	Num of Particles	(Timesteps set to 20)
7.25	5	
11.5	20	
19.75	40	
26	80	
43	160	
102.25	320	
291.25	640	
964.25	1280	
3579.25	2560	
13535.5	5120	
52540	10240	
Threads unset		

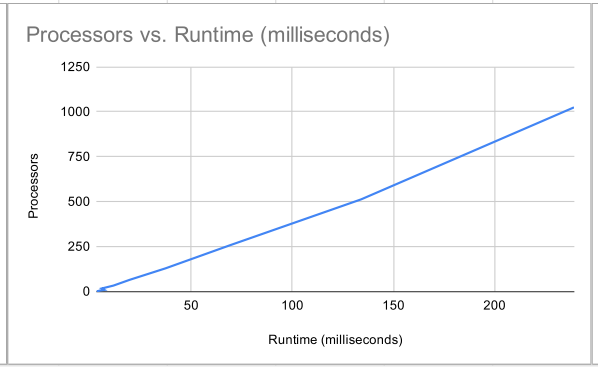
Runtime (millisec	Processors
3.25	1
4.25	2
4.5	4
7	8
5.75	16
11.5	32
19.5	64
37.25	128
69	256
134	512
239.25	1024

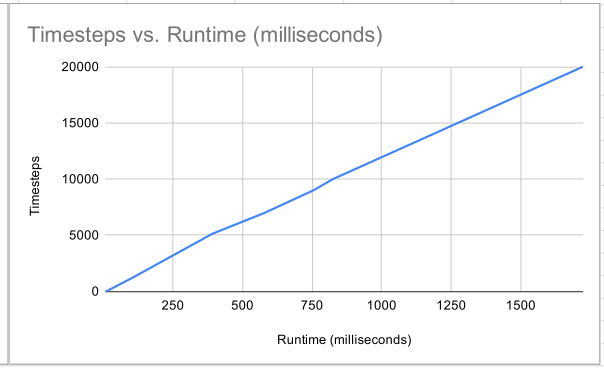
(40 particles at 20 timesteps)
Set with OMP_NUM_THREADS enviornment variable

Runtime (millisec	Timesteps	(Particles set to 20, threads unset)
8.25	5	
11.75	20	
18.25	80	
22.25	160	
36	320	
110	1280	
391.25	5120	
580.5	7000	
755.25	9000	
825.5	10000	
1723	20000	

Taking an average of 4 runs through same settings (num particles, etc.)







From the looks of it, when the number of threads to be set is assigned to OpenMP based on the code (for loops), a linear relationship between timesteps and runtime can be seen. And there seems to be a relationship between the number of particles and the runtime almost resembles a logarithmic function. The runtime with processors set seems to suggest that the overhead cost of creating more threads than required (set with OMP_NUM_THREADS) shows that more threads is not always better.

To run code, compile gifmerge and nBody (with g++ -fopenmp nBody.cpp -o nBody) then run runpsim.sh with ./runpsim.sh <Num. of Particles> <Num. of Timesteps> to visualize the particles.