CS 475/575 -- Spring Quarter 2019

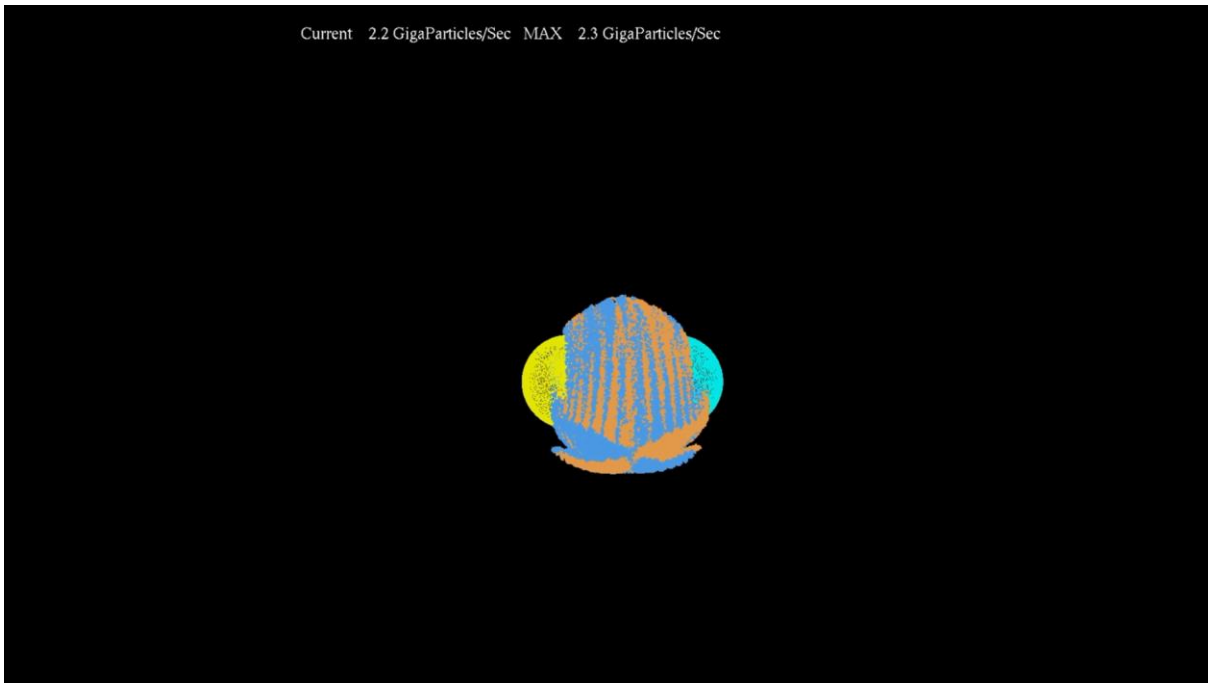
Project #7A

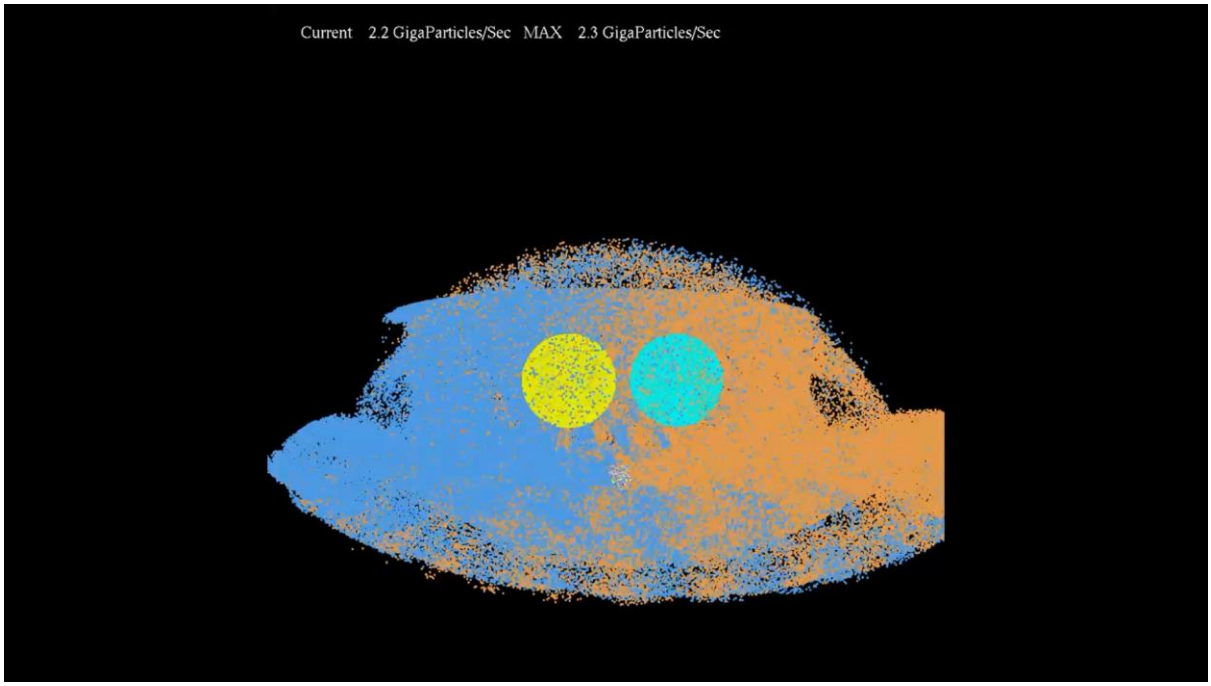
Jiawei Mo
moji@oregonstate.edu

Link: https://media.oregonstate.edu/media/t/0_2b62n7y9 I saw a quality dropping on OSU media, but you can see the patterns.

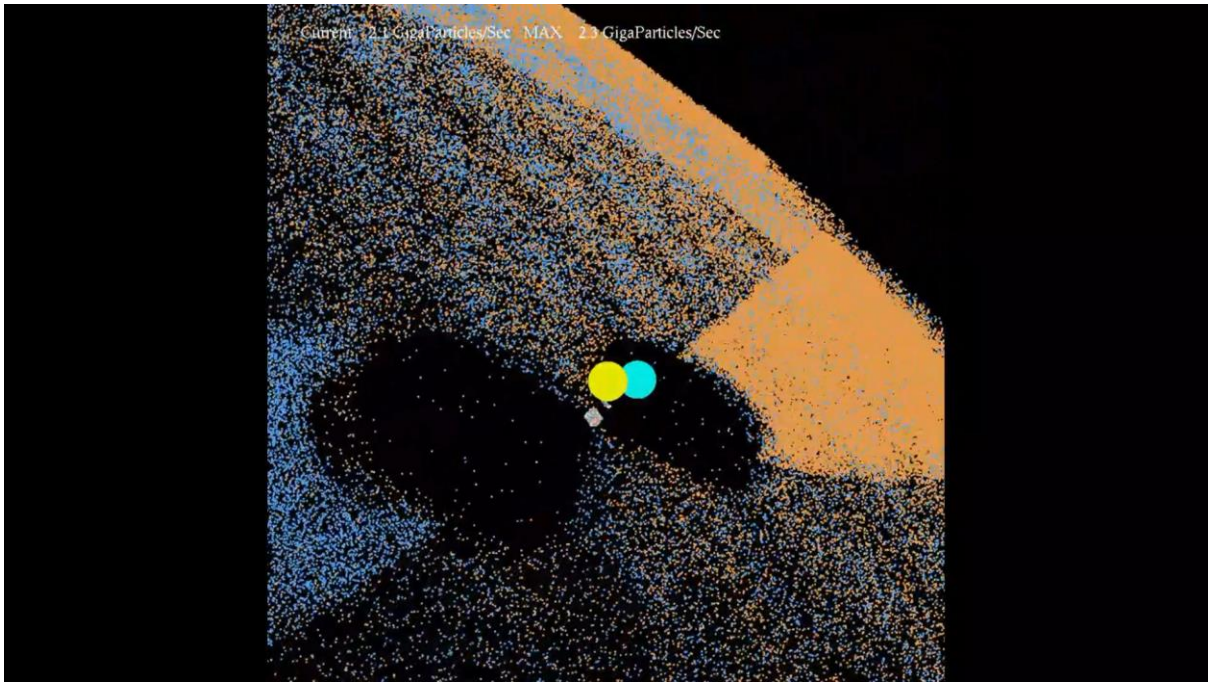
Running on GTX1070

By default, particles will have random colors. If one hits yellow sphere, it turns to orange. If one hits cyan sphere, it turns to blue. Adding oscillation to the acceleration.



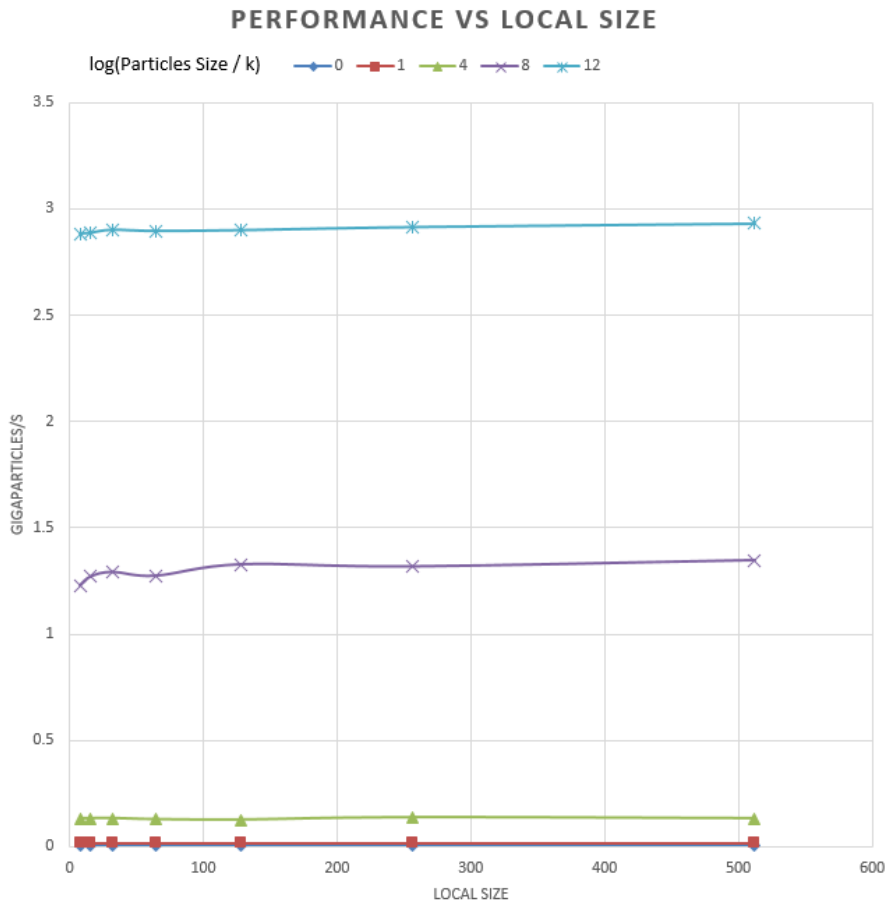

Current 2.2 GigaParticles/Sec MAX 2.3 GigaParticles/Sec

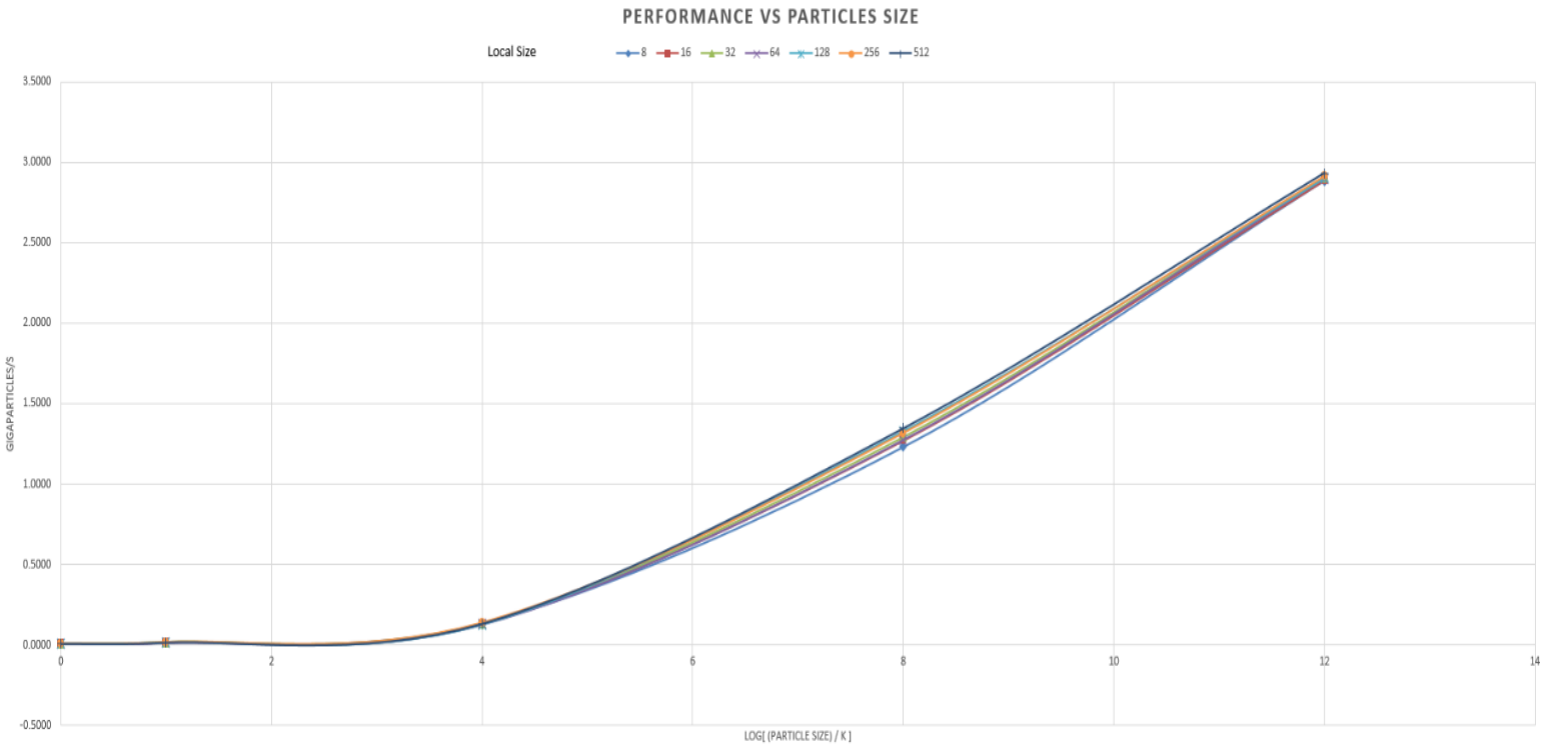

Current 2.1 GigaParticles/Sec MAX 2.3 GigaParticles/Sec



Performance = GigaParticles/s

Particles (2^n * k)	n	8	16	32	64	128	256	512
2^0 * k	0	0.0082	0.0084	0.0085	0.0086	0.0081	0.008	0.0087
2^1 * k	1	0.0165	0.0168	0.0171	0.0167	0.0175	0.0177	0.0166
2^4 * k	4	0.1331	0.1333	0.1337	0.1283	0.1264	0.1382	0.1331
2^8 * k	8	1.2298	1.2705	1.292	1.2741	1.3274	1.3179	1.3468
2^12 * k	12	2.8836	2.8876	2.9024	2.8967	2.9007	2.9156	2.9329





With a fixed local size, the performance rockets up while the particles size increases. This makes sense because a very small size of particles will not have the GPU fully computes their moving. As shown, the sizes around 1k or 2k cannot warm up the GPU.

With a fixed particles size, the performance also increases while an increasing local size. This is a small increasing in the graph since it is GigaParticles/s. By looking at the table, the speed up is around MegaParticles/s. My GPU might not so powerful to have a certain GigaParticles/s speed up. With an increasing local size, it allows more threads that a GPU can manipulate at the same time. That's why the performance is increasing.

In order to a proper use of GPU, I think a large size of computation is needed. And set the local size as big as the GPU allows. These two hints will have the GPU to use as much as threads it has.