Swetha Jayapathy
Student ID : 934041047
Instructor : Mike Bailey
CS575: Introduction to Parallel Programming
June 8, 2020

Project #7a

1. A web link to the video showing your program in action

   https://media.oregonstate.edu/media/t/0_1gac9dmv

   If the above video is not clear, kindly see the below one

   https://media.oregonstate.edu/media/t/0_1wjkcops

2. What machine you ran this on

   I ran my code on my laptop – Lenovo – L390 Yoga using Visual Studio

   Operating System - Windows 10

   CPU – Intel core i5 processor

   Memory – 8 GB RAM

```
Name    = 'Intel(R) OpenCL HD Graphics'
Vendor  = 'Intel(R) Corporation'
Version = 'OpenCL 2.1 '
Profile = 'FULL_PROFILE'
Number of Devices = 1
Device #0:
        Type = 0x0004 = CL_DEVICE_TYPE_GPU
        Device Vendor ID = 0x8086 (Intel)
        Device Maximum Compute Units = 24
        Device Maximum Work Item Dimensions = 3
        Device Maximum Work Item Sizes = 256 x 256 x 256
        Device Maximum Work Group Size = 256
        Device Maximum Clock Frequency = 1100 MHz
```

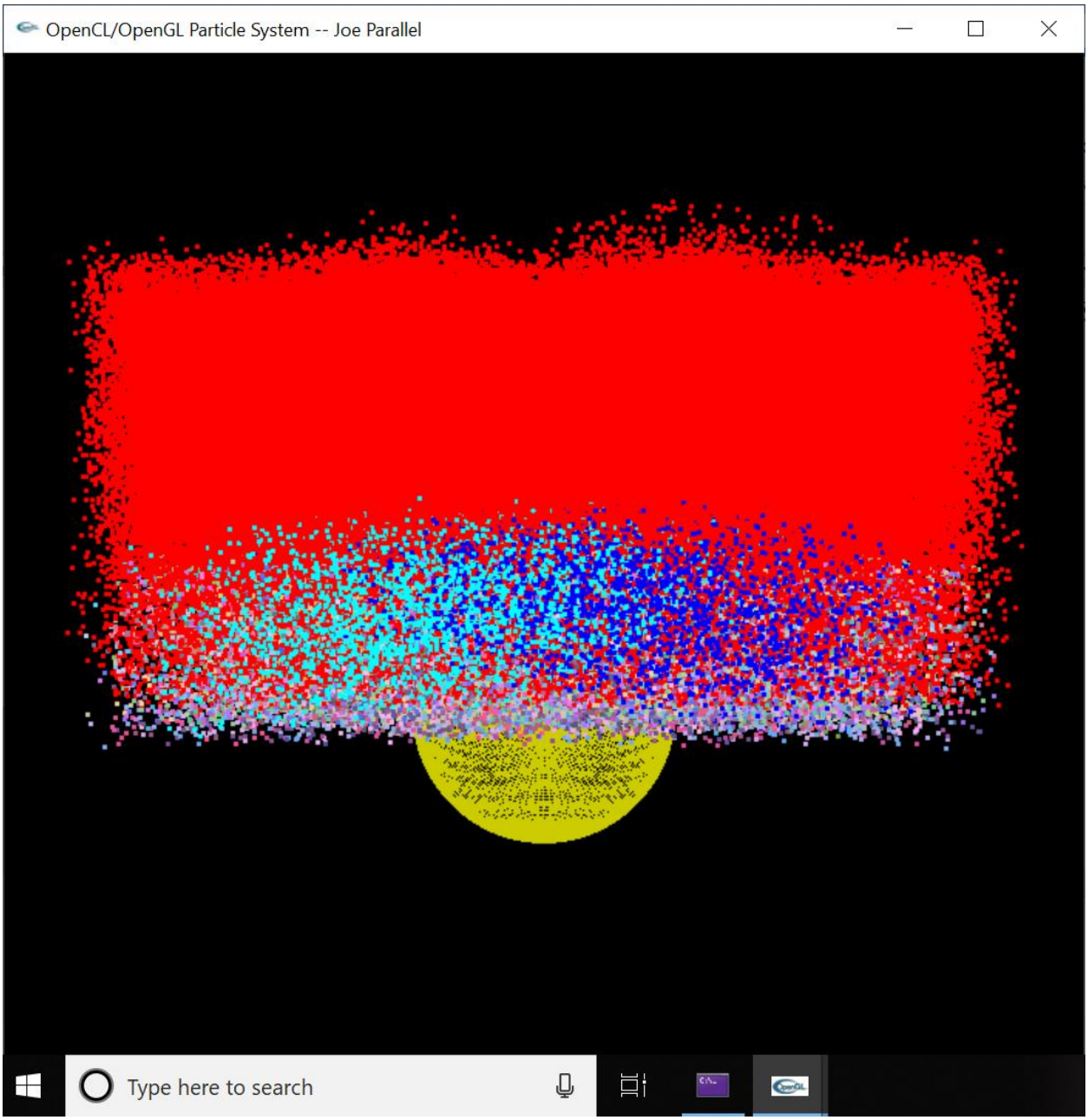
3. What predictable dynamic thing did you do with the particle colors (random changes are not good enough)

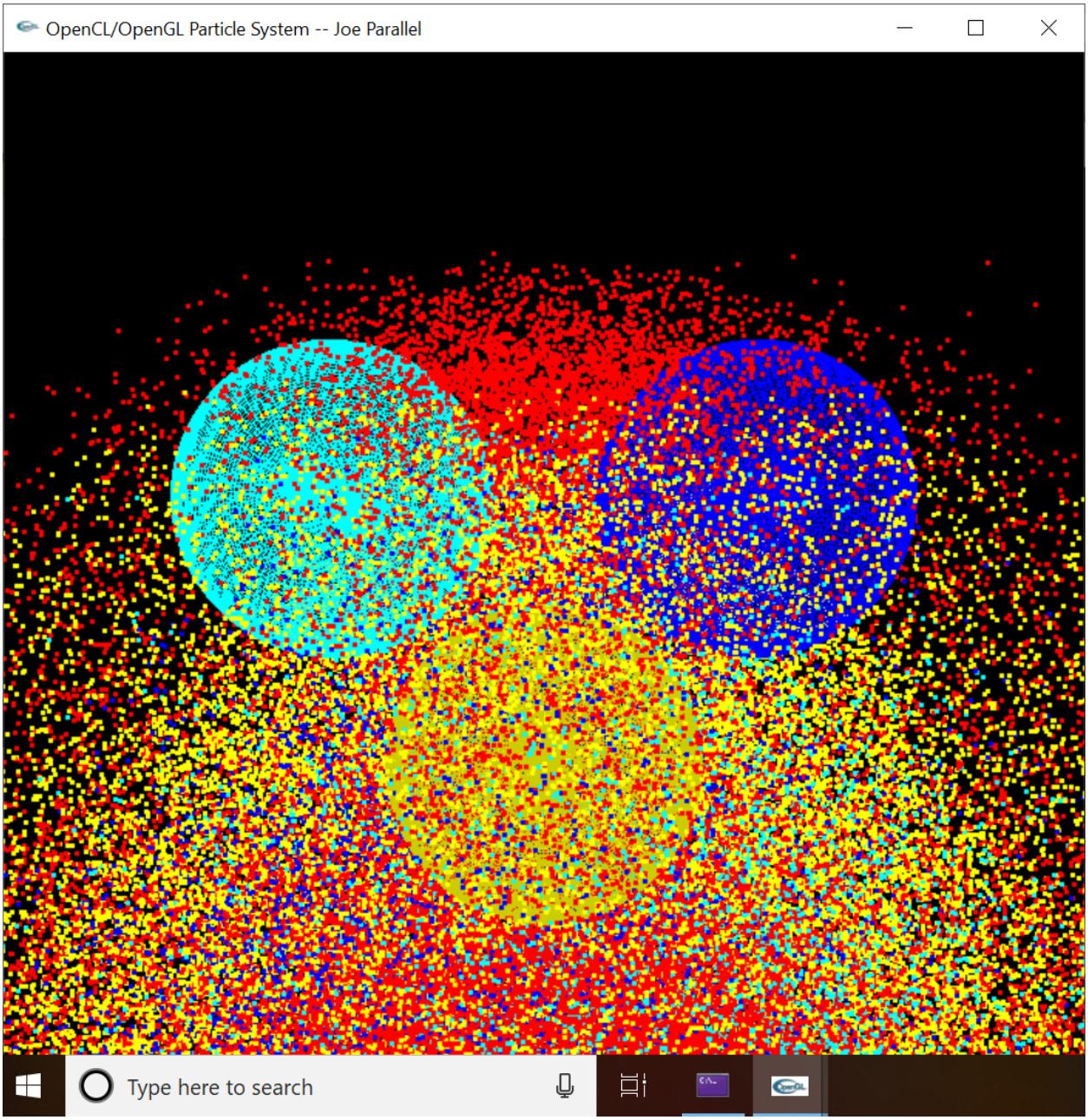
- I have included 2 additional spheres and hence there are total 3 Spheres of colors – Cyan, Blue and Yellow.
- When the particles are in positive Y axis (above the X axis), I give them red colors.

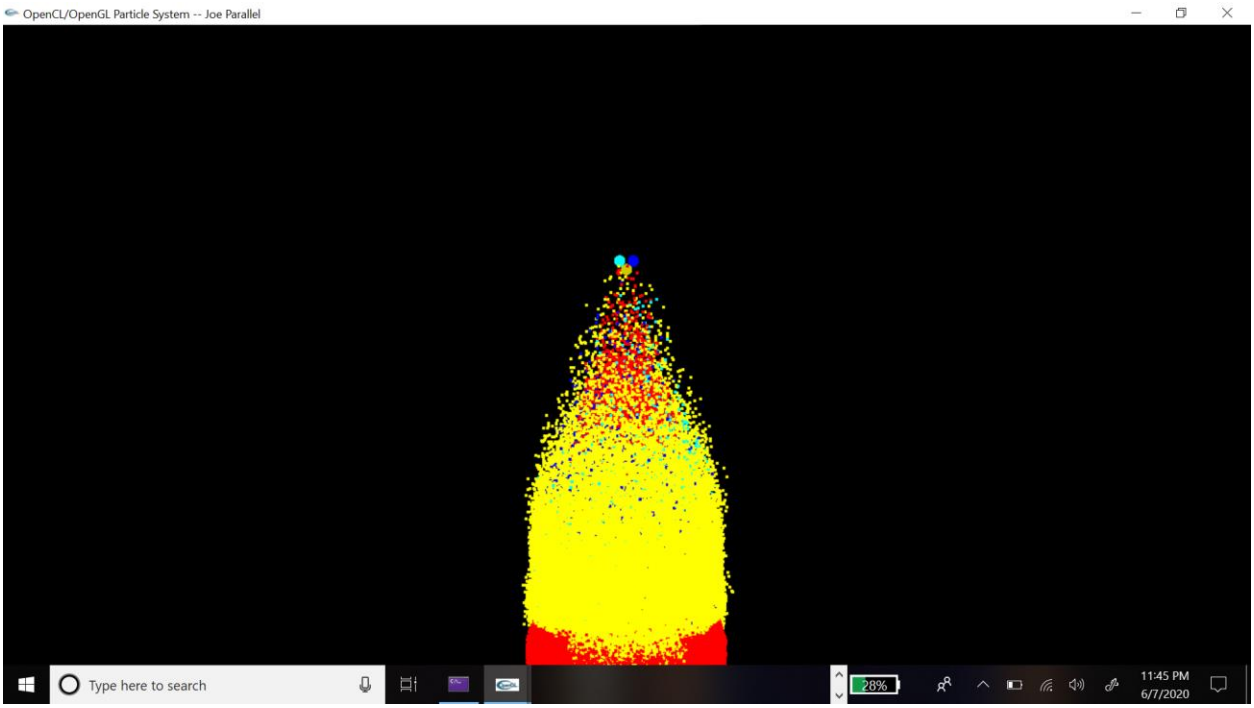
- When the particles are bouncing from the spheres, the color of the particles will be the respective color of the spheres.

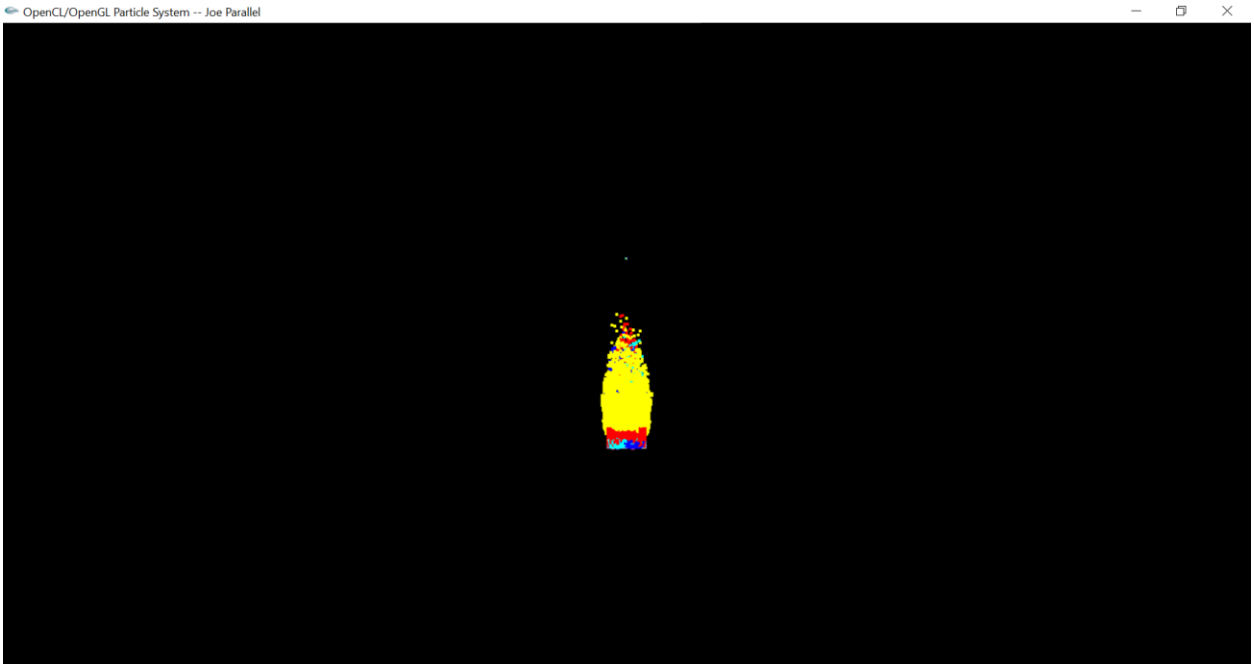
4. Include at least one screen capture image of your project in action

The below screen shot is for Number of Particles with 1024 * 1024



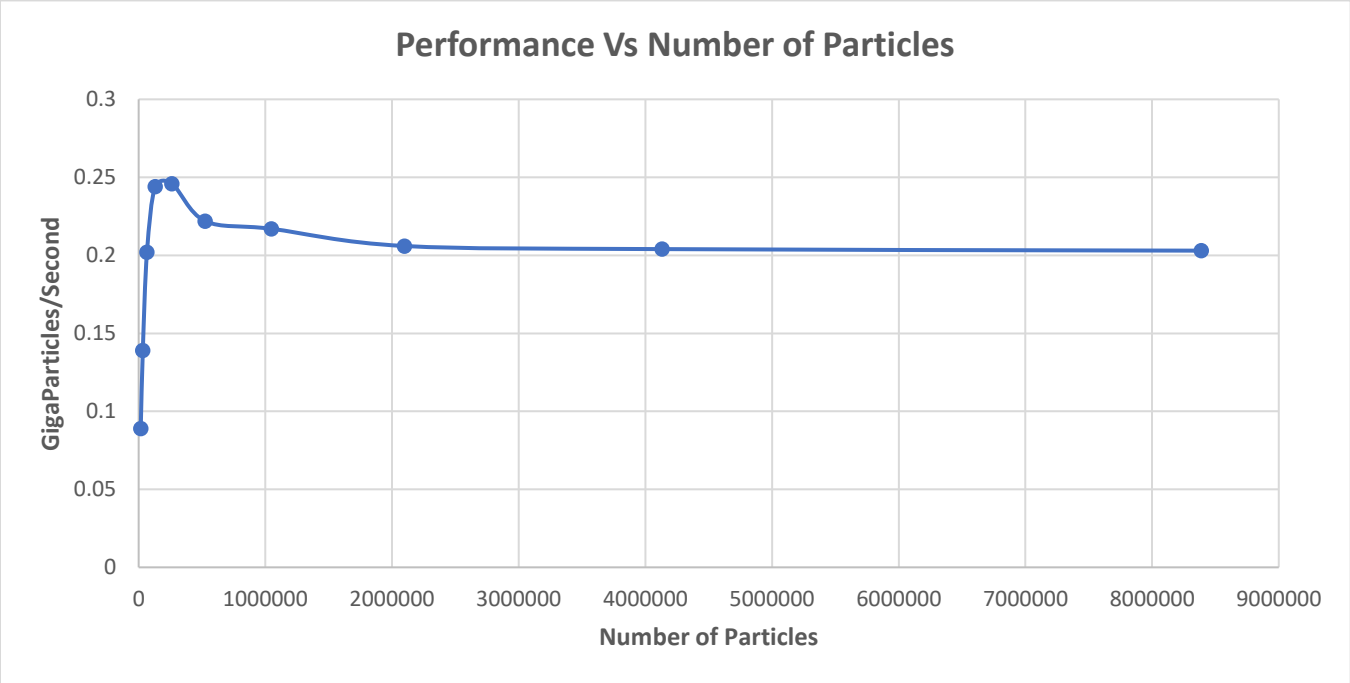

OpenCL/OpenGL Particle System -- Joe Parallel
Type here to search


OpenCL/OpenGL Particle System -- Joe Parallel
Type here to search
28%
11:45 PM
6/7/2020


OpenCL/OpenGL Particle System -- Joe Parallel

5. Show the table and graph

Number of Particles	Performance(GigaParticles/Sec)
16384	0.089
32768	0.139
65536	0.202
131072	0.244
262144	0.246
524288	0.222
1048576	0.217
2097152	0.206
4132864	0.204
8388608	0.203



6. What patterns are you seeing in the performance curve?

We could see from the graph that for lower data size (number of particles) we get a low performance, when the data size reaches 65536, we could see that the performance gets better. We could also see that there is a gradual increase in the performance with the increase in the number of particles. Although the performance hikes when number of particles is 131072, thereafter the performance flats out (almost same) from 262144. But ideally, the performance gets better for higher data size.

7. Why do you think the patterns look this way?

With low values of the data size, the GPU doesn't get the minimum load required to process and hence the low performance. The gradual increase in the performance with respect to the increase in the Number of particles is the efficiency of the GPU to process huge amounts of data parallelly. This is because the GPU is occupied to its fullest capacity as the data size gets bigger. The more the data size is, the efficient use of work items in each work group. GPU gives us the high performance with respect to Parallelism.

8. What does that mean for the proper use of GPU parallel computing?

As we could see from the above observation, for the proper use of the GPU, its full capacity is to be utilized. When the Number of particles is less, it is not properly using the GPU efficiently, as the Number of particles increases, the GPU's efficiency also increases as the minimum warp size is satisfied(32).
The more the data size, we could keep the GPU busy. Moreover, we see that for higher level of parallelism GPU is best than CPU with respect to performance. This is the reason GPU is best for Gaming or a simulation as it has to get everything done at the 60^{th} of the second and also update the screen in time. This shows how OpenCL/OpenGL interoperability is highly productive.