

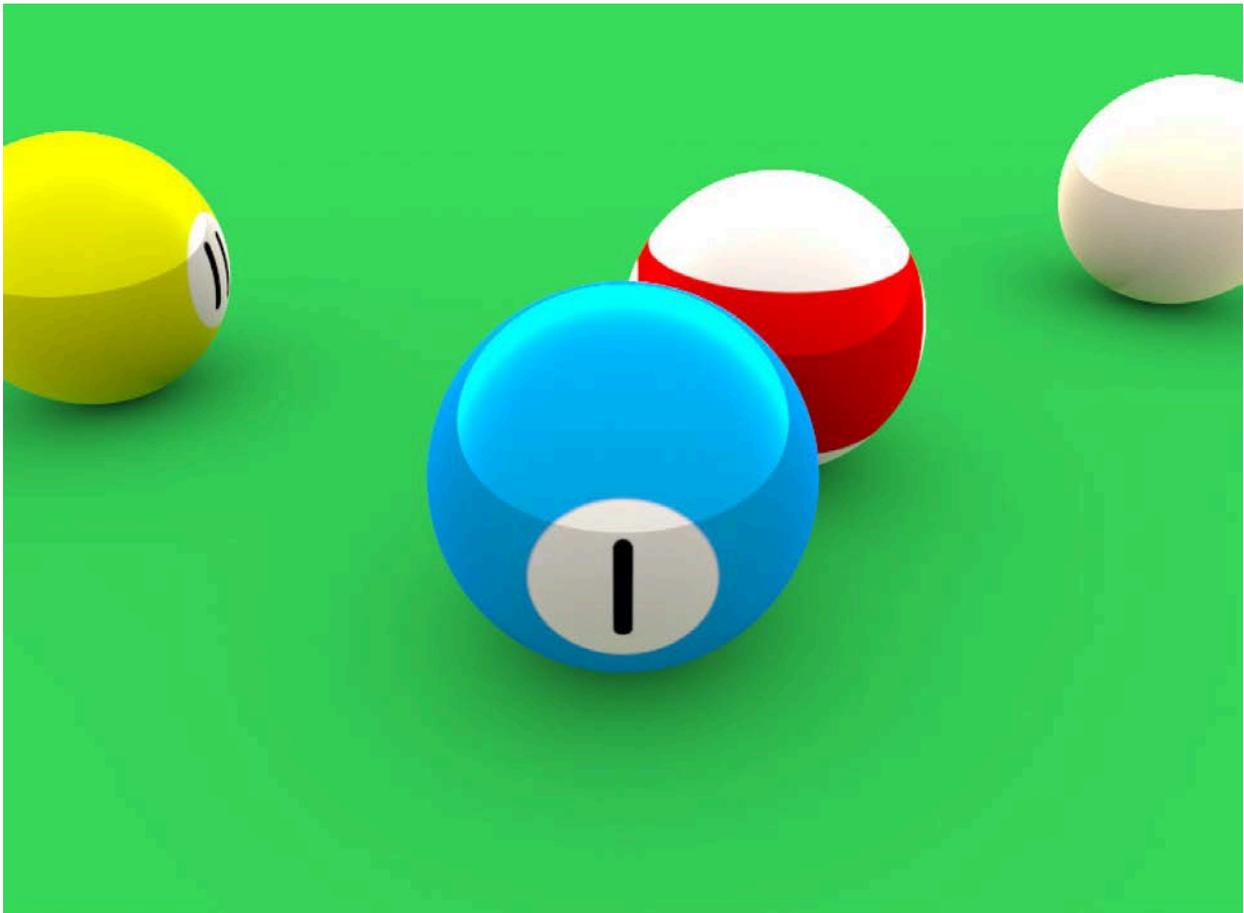
Parallel Computer Architecture and Programming

Project Dracuda Checkpoint Report

Summary of Work completed so far

We have implemented a GPU ray tracer in CUDA. The ray tracer currently supports various effects such as fresnel reflection, ambient occlusion, diffuse interreflection, environment mapping and custom materials which enable the use of mathematically represented fonts.

We would want to implement a pool game as an interactive demo, so we created a demo scene to test the ray tracer's functionalities. The image rendered is shown below.



Refined goals and deliverables

Although we are behind our schedule so far, we still believe we will be able to produce all the deliverables on time. Our goals for the Parallelism Competition remain the same:

We will develop an interactive pool game to show the possibility of playing game with amazing graphics in real-time.

We will implement the load balancer algorithm to coordinate the worker nodes to reduce the overall latency.

We will further polish the GPU ray tracer to increase its rendering speed, and support more effects if necessary.

Issues and concerns

Our ray tracer is able to render the image above with 800x600 resolution and 2500 rays/pixel in 15 seconds. However, this performance is still not satisfying enough for a real-time experience. We will first achieve real-time rendering with a reasonable image quality (for example 100 rays/pixel), and then gradually optimize and improve the quality.