Karshan Arjun
June 27, 2018

Programming Massive Parallel Systems Project 2 Report

My first way of speeding up the program was to generate the data by using cuda's library curand_kernel.h which helped generate data faster with the curandState and the curand_init to create a double with was using the clock as a variable to gather numbers which were unique, I created the generate_data function.

```
__global__ void
generate_data(atom *a, long long a_num){
  int i = threadIdx.x + (blockIdx.x * blockDim.x);

  // Create random numbers, using clock so no two values can ever be the same
  curandState state;
  curand_init(((unsigned long long)clock() + i) * BOX_SIZE, RAND_MAX, 1, &state);
  a[i].x_pos = curand_uniform_double(&state) * BOX_SIZE;
  a[i].y_pos = curand_uniform_double(&state) * BOX_SIZE;
  a[i].z_pos = curand_uniform_double(&state) * BOX_SIZE;

  // Using Barrier Synchronization
  __syncthreads();
}
```

My second approach to for making my project one code more optimize was to now use a 2D approach on the function PHD_Baseline, instead of a 1D approach like what I did on the first project this was a more faster way to get the p2p distance. I added the j variable to use the block dimension y and the thread index y.

```
__global__ void
PDH_baseline(bucket *histogram, atom *atom, double weight) {
   int i = (blockIdx.x * blockDim.x) + threadIdx.x;
   int j = (blockIdx.y * blockDim.y) + threadIdx.y;

   // Get the distance and then the position of the histogram with 1
   if (i < j) {
      double distance = p2p_distance(atom, i, j);
      int position = (int) (distance / weight);
      histogram[position].d_cnt++;
      // Using Barrier Synchronization
      __syncthreads();
   }
}
```

Project one running time:

```
Time to generate on the Kernal with GPU: 0.05581 ms
Running time for CPU version: 2.303566
Running time for GPU version: 0.000155
Total distance for CPU version: 49995000
Total distance for GPU version: 49995000
```

Project two running time:

```
Time to generate on the Kernal with GPU: 0.01754 ms
Running time for CPU version: 2.307015
Running time for GPU version: 0.000090
Total distance for CPU version: 49995000
Total distance for GPU version: 49995000
```