# UPC Notes

Eric Andrews

April 22, 2018

# UPC Implementation

## 1 General Structure

Particles and bins become shared across all threads. Each step is done with affinity to the particle or bin being operated on; anywhere a bin or a particle is being changed is locked to avoid data races. Particle initialization and statistical data are handled by thread 0.

## 2 Reduction

I made some attempts at a reduction using shared arrays of ints and doubles instead of the single variables in the serial code. Thread 0 was supposed to then sum these up and report the final statistics. I was unable to get this working; my attempt, however, is included in the file upc-experimental.c.

## 3 Blocking

For code of legibility, I held off on blocking until I had finished the rest of my code. A few forays into it with my partially functional code could not be convinced to work; moreover, I am unconvinced that it would have a significant effect due to the high communication cost of this simulation on UPC.

## 4 Known Issues

- Not all particles interact; even when running with 1 thread, some particle is not interacting.

- The reduction is non-functional.

## 5 Performance

| **Threads** | **Time** |
|---|---|
| 1 | .996 |
| 2 | .851 |
| 4 | .552 |
| 8 | .363 |

There is a wide variance in runtime between runs in which all particles interact and runs in which they do not. Likely there is a thread-specific condition which triggers a premature termination of the program or skipping over multiple particles. I was not able to find where this occurs.

Overall, the performance increase is not ideal, but not insignificant; in runs for which the particles interacted properly, there was linear speedup for increasing the number of threads used.

## 6 Attempted Methods

Most of what I tried consisted of changing what was shared and debugging the fallout from that. The main two lines of attack I used were passing ints vs particle_t's to array_t_get. I finally went with ints because it meant less negotiation with shared space; using the particle array already in shared space proved simpler and more intuitive.