

COSC3500

2D Orbital Simulation Report

Maxwell Bo (43926871)

August 17, 2018

Description

Implementation

Correctness

Performance & Scaling

The addition of the `-march=native` compiler flag, which enables the use of all CPU specific instructions, provided no recognizable improvement in running time.

The use of the GCC and Clang Profile-Guided Optimisation features provided no recognizable improvement in running time.

Distressingly, `-O0`, `-O1`, `-O2`, `-O3` showed no recognizable improvement in running time. `-Ofast` led to an -4%-ish performance regression.

Incl.	Self	Called	Function	Location
100.00	0.00	(0)	0x0000000000001030	ld-2.17.so
100.00	0.00	2	_dl_runtime_resolve_xsave	ld-2.17.so
100.00	0.00	1	0x00000000000040df6	nbody
100.00	0.00	(0)	(below main)	libc-2.17.so
100.00	9.33	1	main	nbody: main.cpp, basic_string.h, string_conversio...
89.22	26.33	672 000 056	Body::exert_force_unidirec...	nbody: Body.cpp
62.90	2.93	672 033 712	distance(double, double, ...	nbody: utils.cpp
59.97	11.70	672 033 711	hypot	libm-2.17.so
48.27	48.27	672 033 712	__hypot_finite	libm-2.17.so
0.63	0.63	48 000 000	Body::frog(double)	nbody: Body.cpp
0.52	0.52	48 000 008	Body::leap(double)	nbody: Body.cpp
0.21	0.21	48 000 000	Body::reset_force()	nbody: Body.cpp
0.07	0.00	601	dump_timestep(double, st...	nbody: main.cpp, stl_vector.h, stl_iterator.h

Two performance fixes were devised.

```
void Body::exert_force_unidirectionally(const Body* there) { double m1 = m; double m2 = there->m; double r = distance(*this, *there);  
double F = (G * m1 * m2) / r2;  
double delta_x = there->x - x; double delta_y = there->y - y;  
// turn the displacement vector between our two points into a force vector // of the desired  
magnitude double scale_factor = F/r;  
Fx += delta_x * scale_factor; Fy += delta_y * scale_factor;  
Here, we are recalculating  
gcc -g -lstdc++ -Wall -pedantic -Wextra -std=c++11 -lm -O3 -march=native Body.o QuadTree.o  
main.o utils.o -o nbody Barnes-Hut enabled: false Leapfrog enabled: true Total CPU time was  
13.938825 12000001 simulation steps computed
```