COSC3500 Forum Presentation

Max Bo

22nd of October

Outline

Description

The task was to create a stock-standard, 2-dimensional gravitational *n*-body simulator.

All bodies were to be assumed to be point masses. The simulation was to be accurate, maintaining a constant total energy, and exhibiting phenomena such as apsidal precession.

Demo

Integration I

$$F = G\frac{m_1 m_2}{r^2}$$

$$a_i = F(x_i)$$
$$v_{i+1} = v_i + a_i \Delta t$$

Integration II

Dehen and Read note that the Euler method 'performs very poorly in practice', further noting that 'errors are proportional to Δt^2'. They contrast it with the second-order *Leapfrog* symplectic integrator, which is 'heavily used in collisionless N-body applications'.

$$
\begin{aligned}
x_i &= x_{i-1} + v_{i-1/2}\,\Delta t \\
a_i &= F(x_i) \\
v_{i+1/2} &= v_{i-1/2} + a_i\,\Delta t
\end{aligned}
$$

which only requires a single acceleration calculation per every two half timesteps

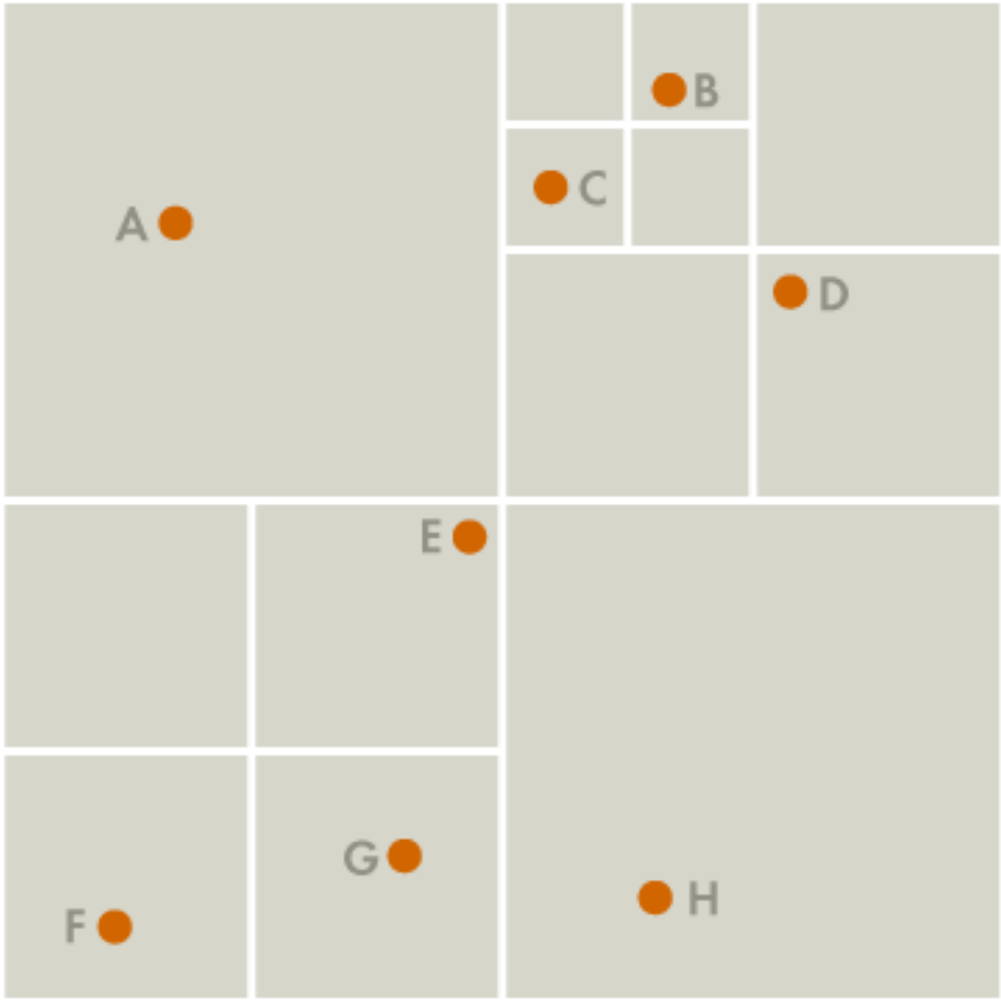
Integration III

and a 'kick-drift-kick' form

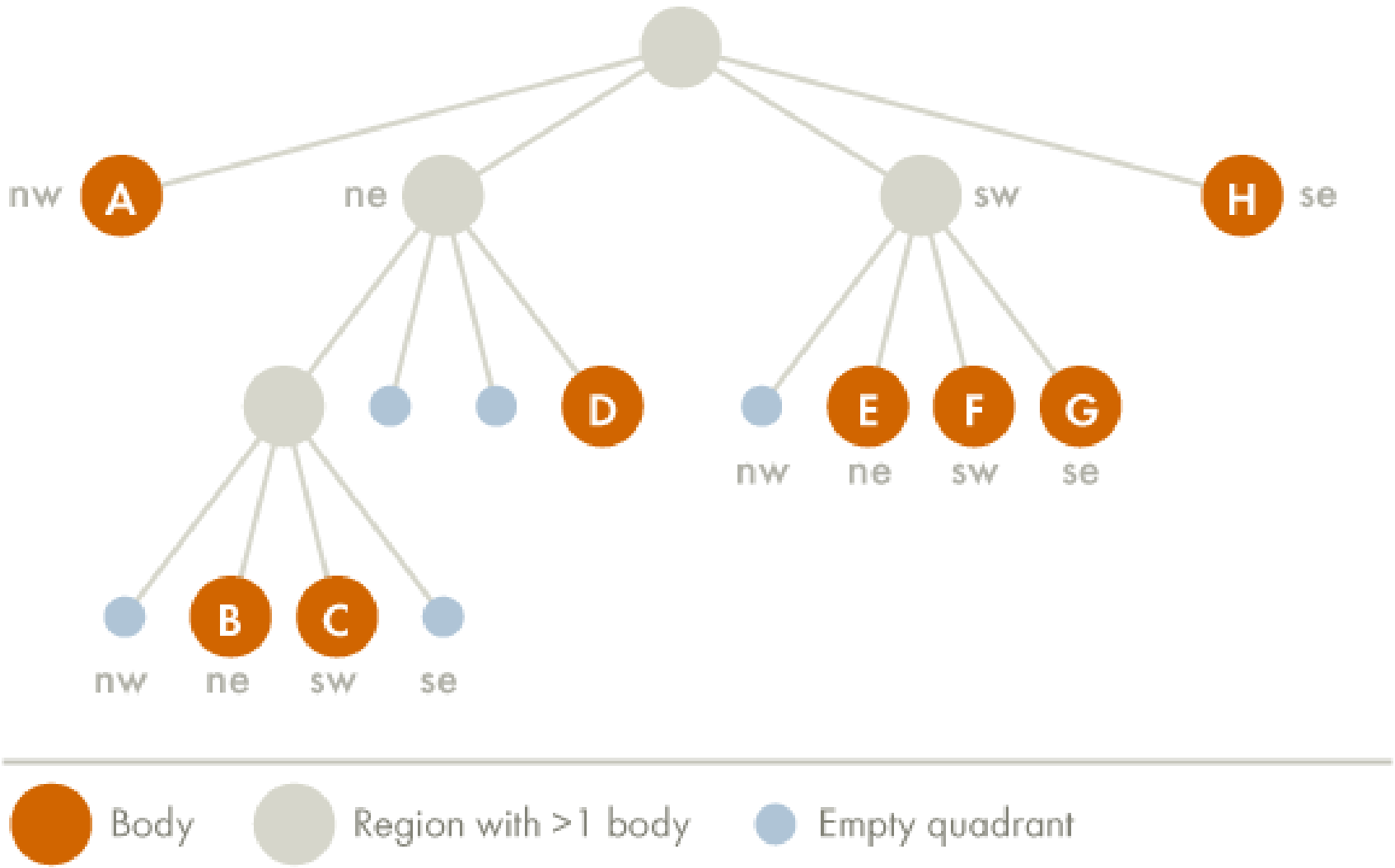
$$v_{i+1/2} = v_i + a_i \frac{\Delta t}{2}$$
$$x_{i+1} = x_i + v_{i+1/2} \Delta t$$
$$v_{i+1} = v_{i+1/2} + a_{i+1} \frac{\Delta t}{2}$$

that is stable with variable timstepping, but incurs an additional acceleration calculation per every two half timesteps.

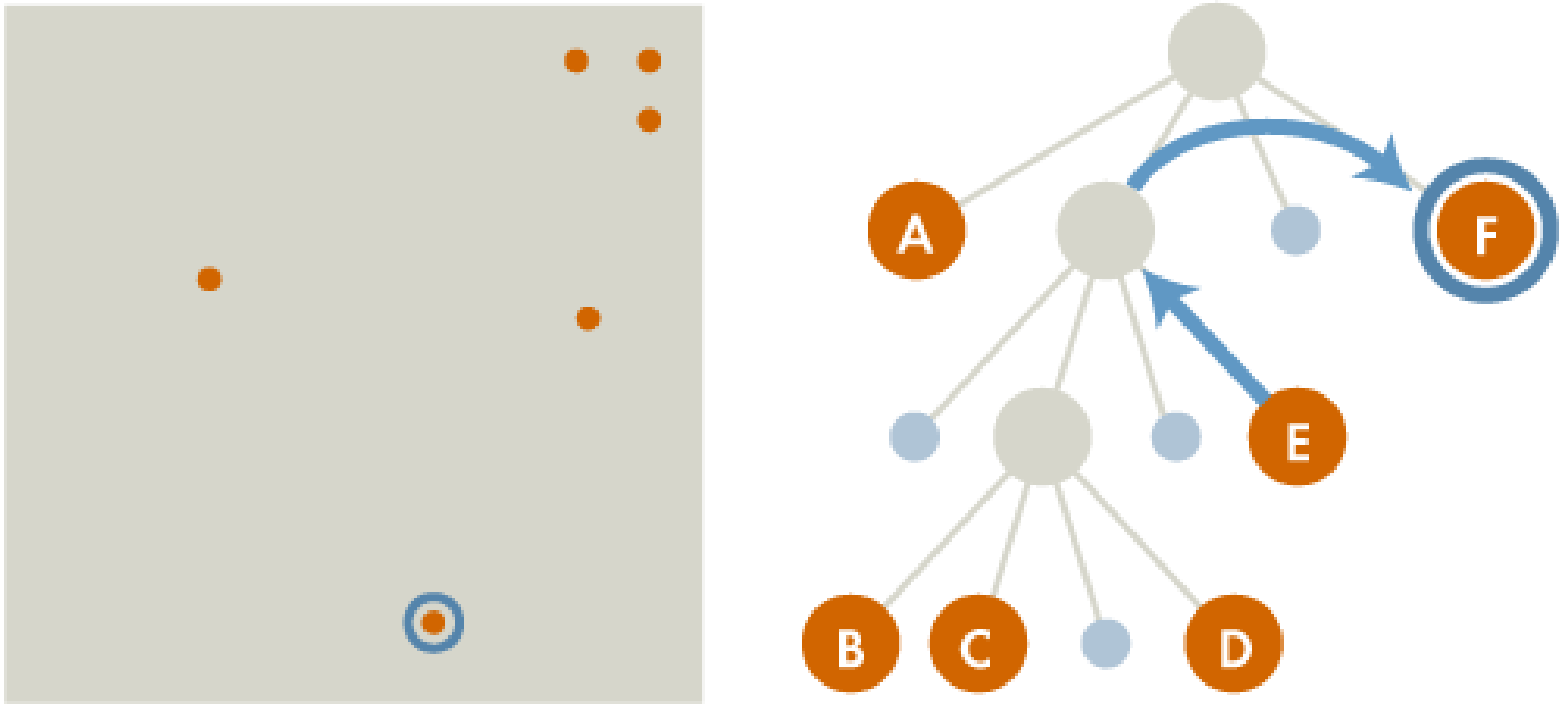
Barnes-Hut I



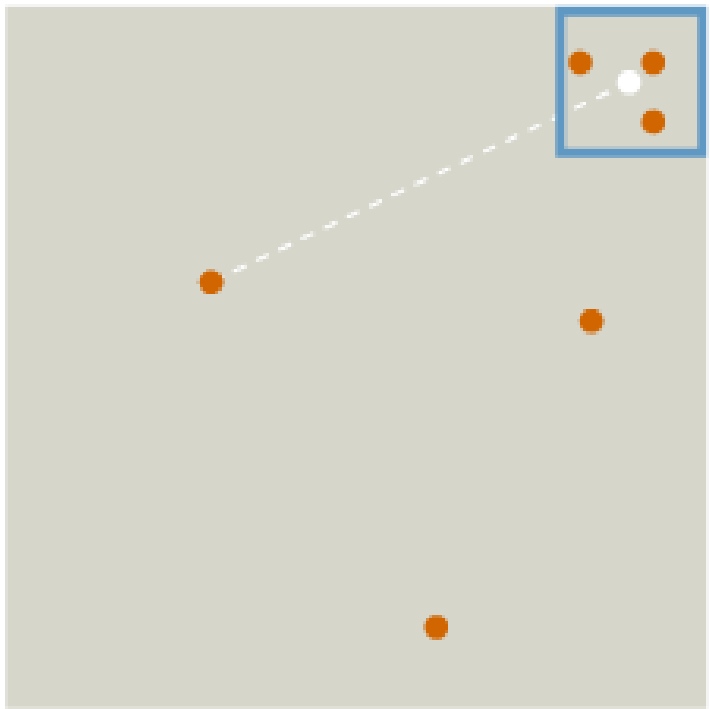
Barnes-Hut II

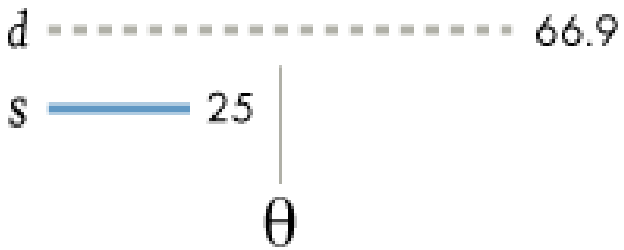


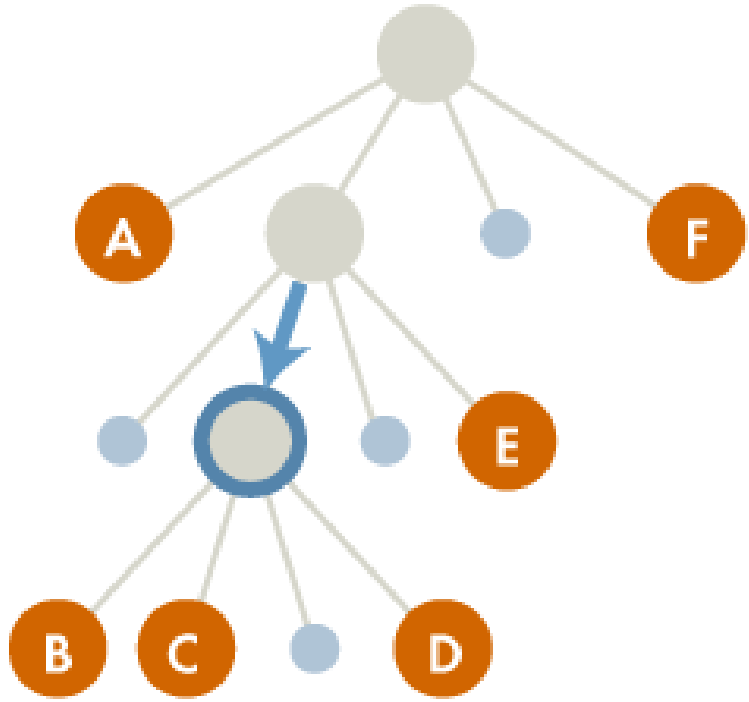
Barnes-Hut III



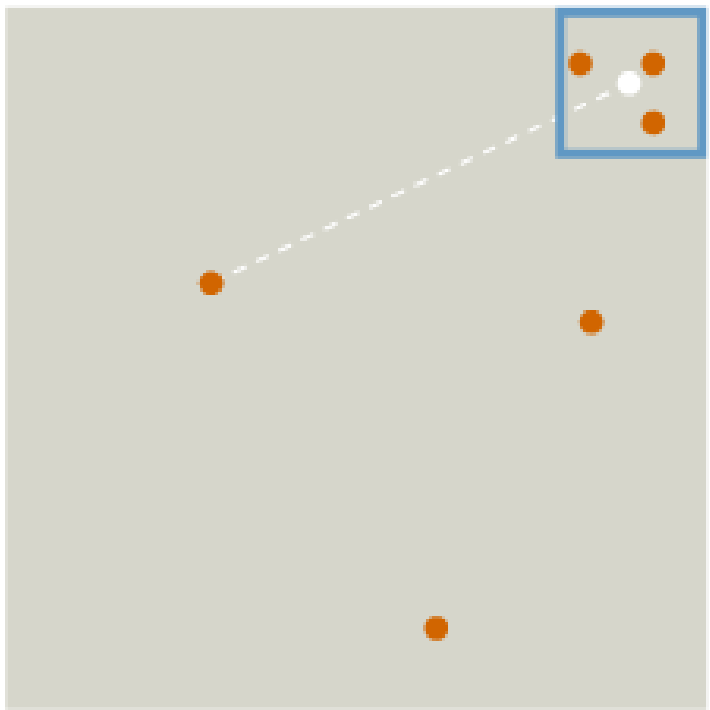
Barnes-Hut IV

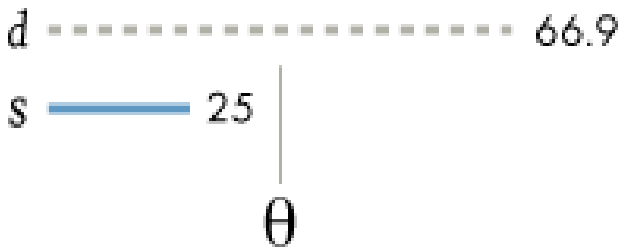


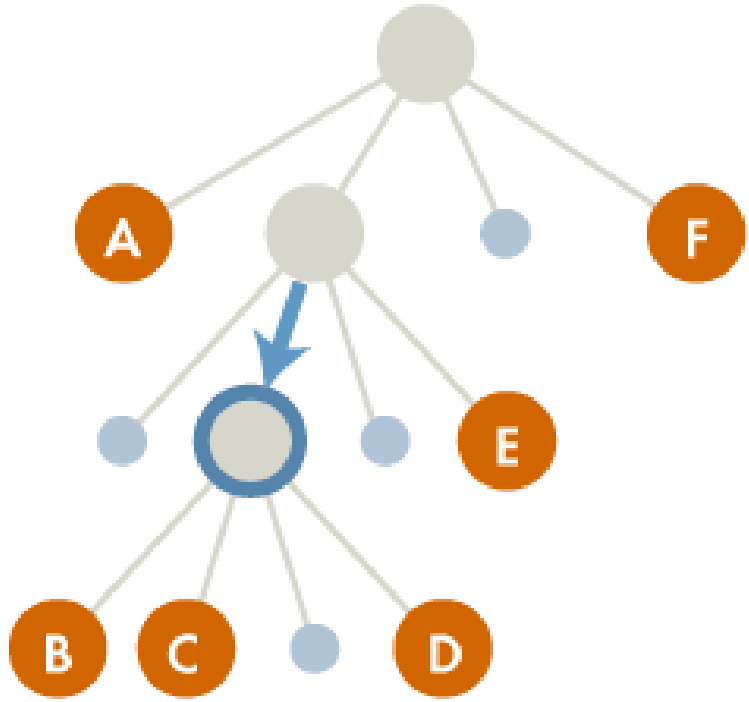




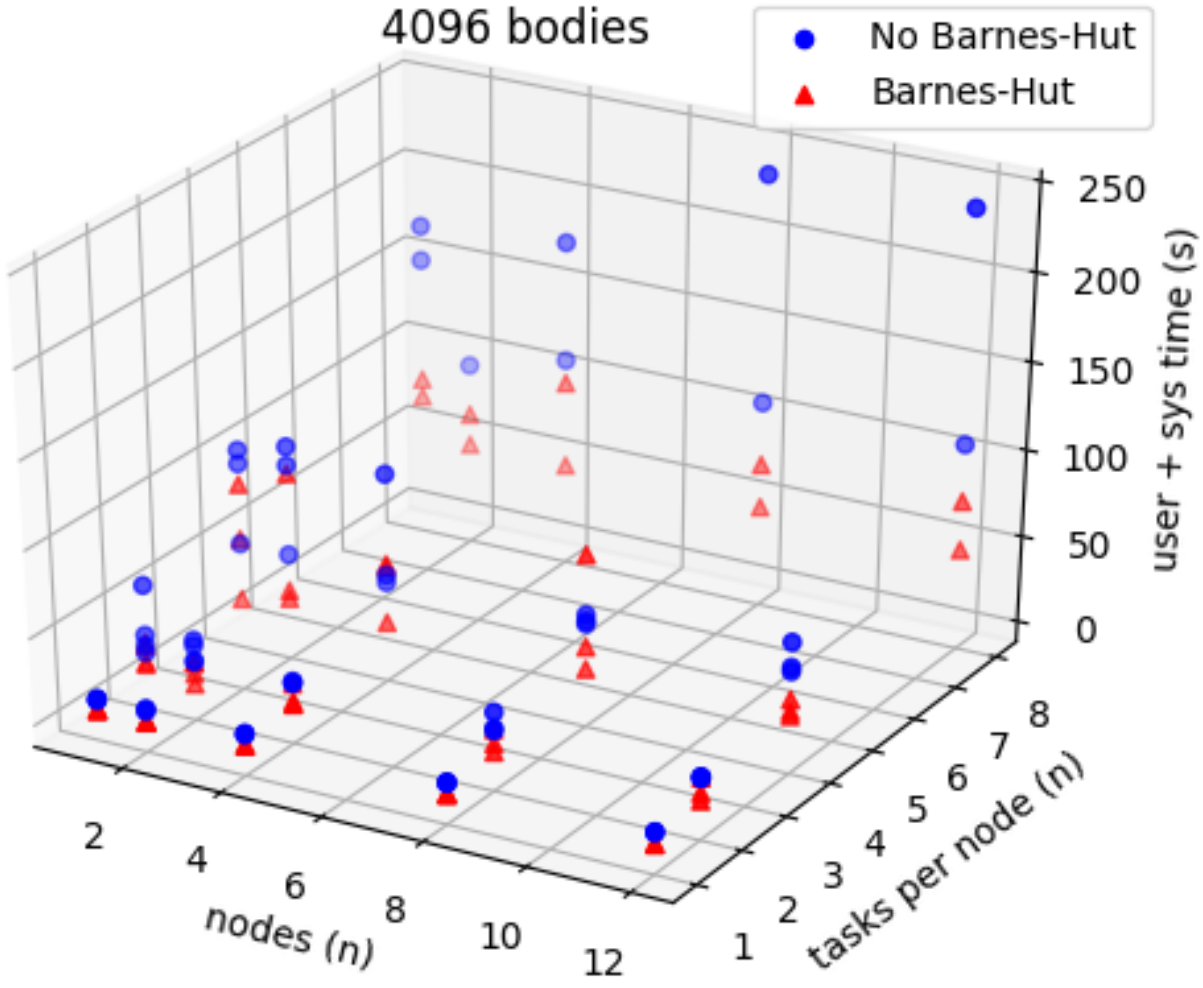
Barnes-Hut V



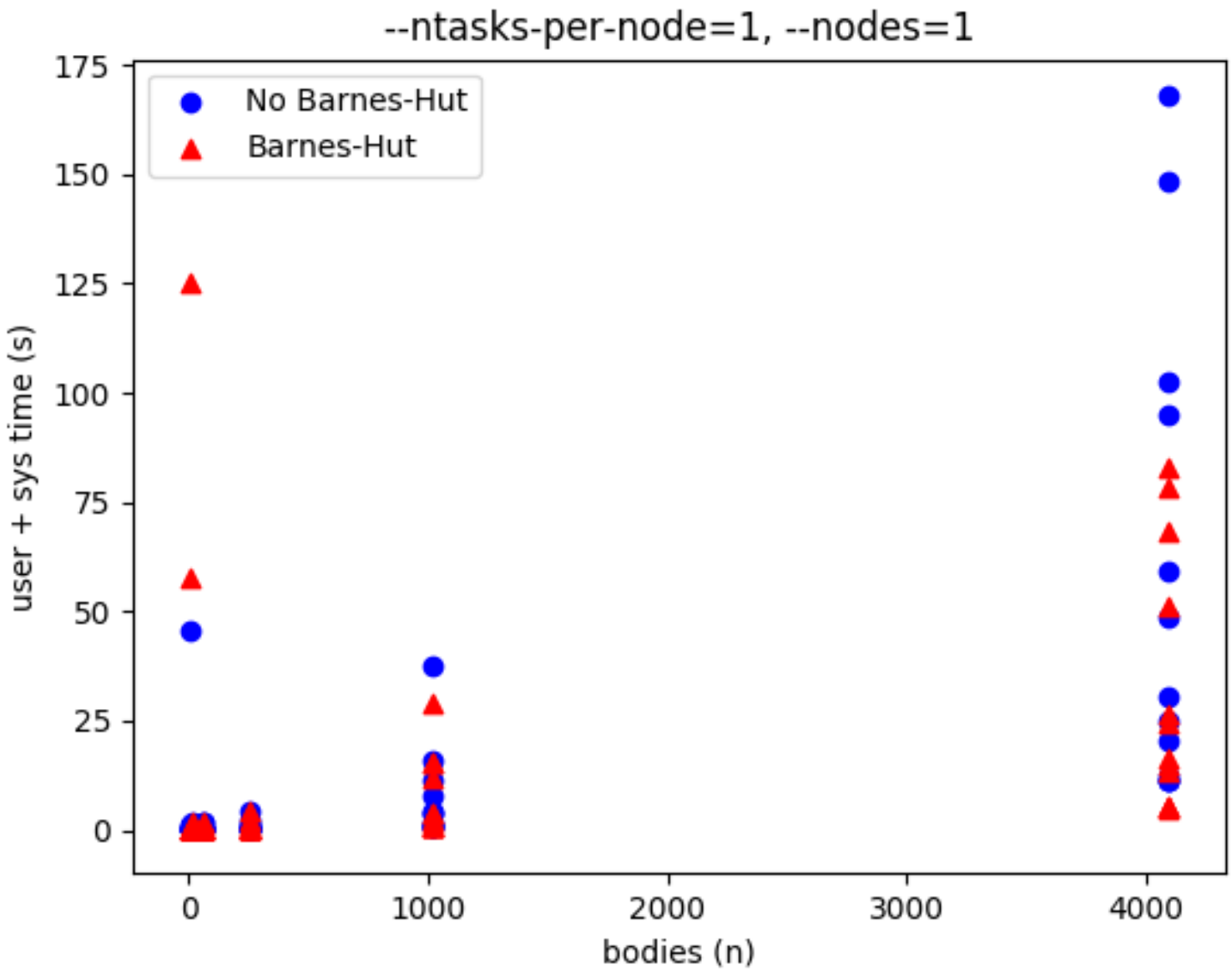




Barnes-Hut VI



Barnes-Hut VII

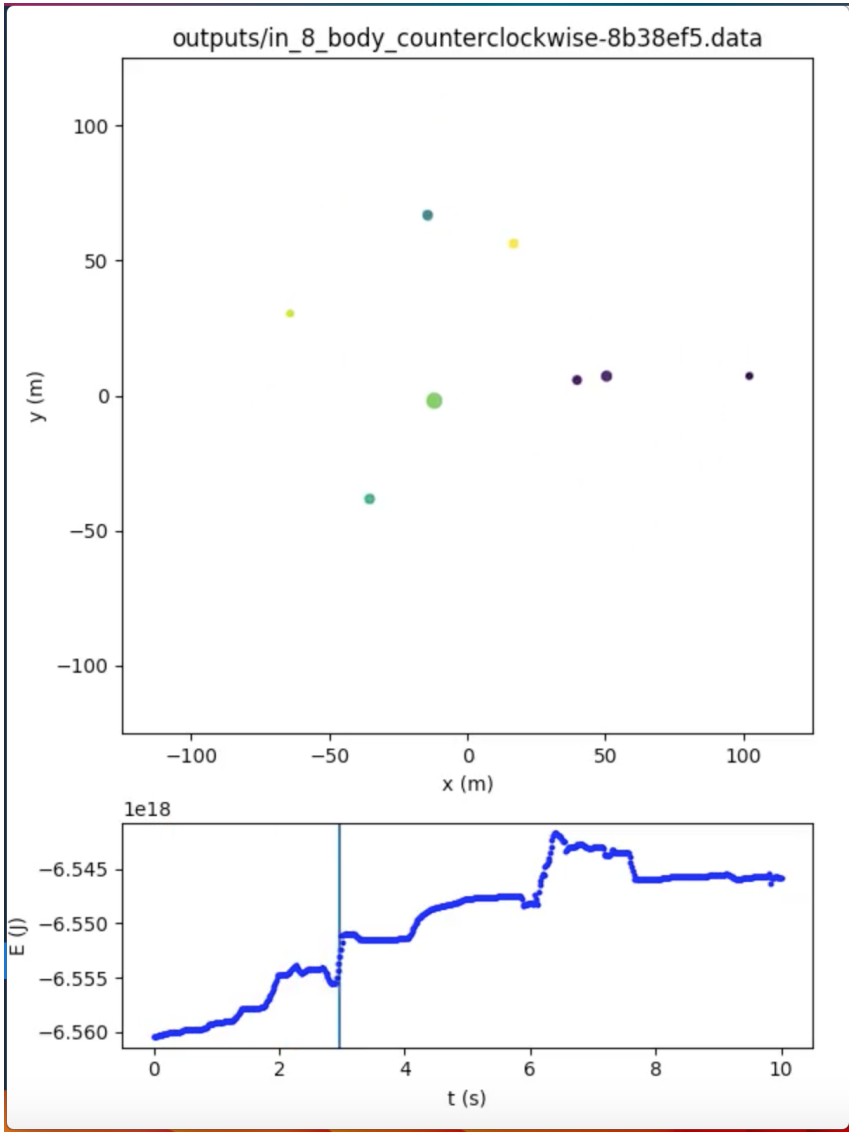


Correctness I

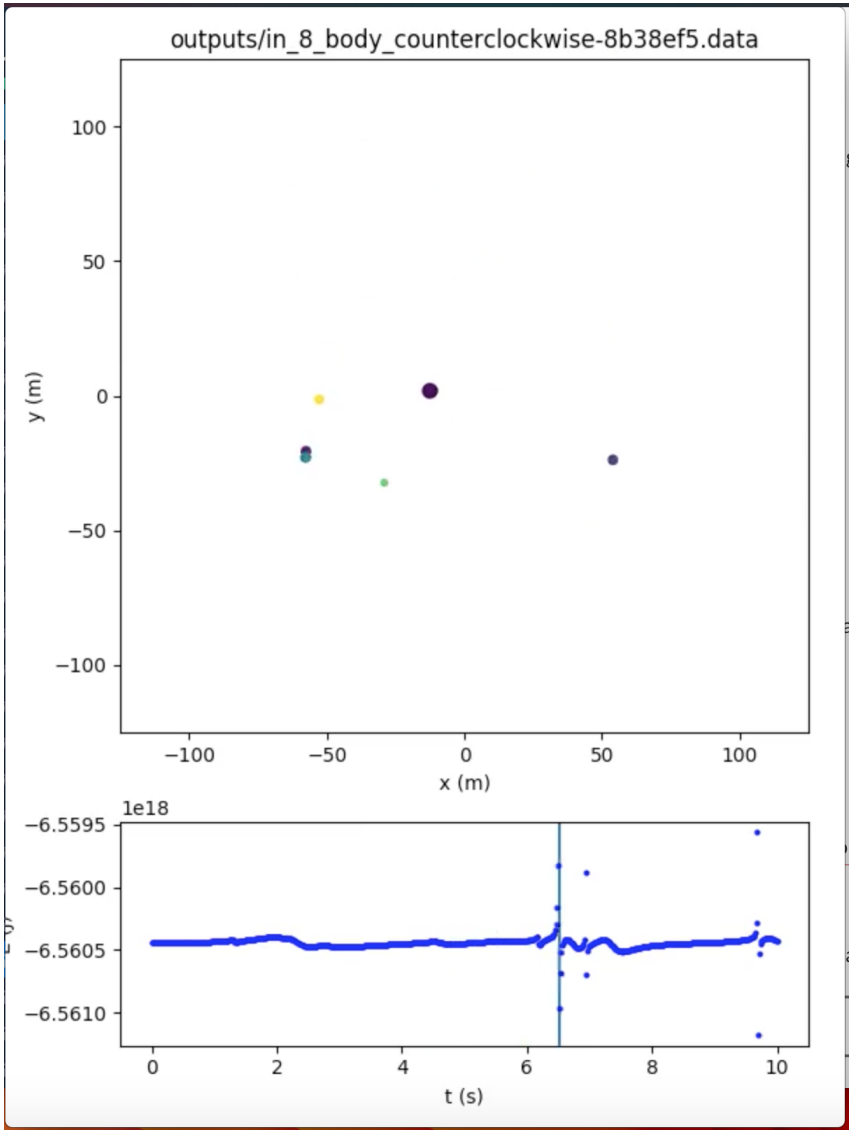
$$U = -G\frac{mM}{R}$$

$$E_{\mathrm{k}} = \frac{1}{2}mv^2$$

Correctness II



Correctness III

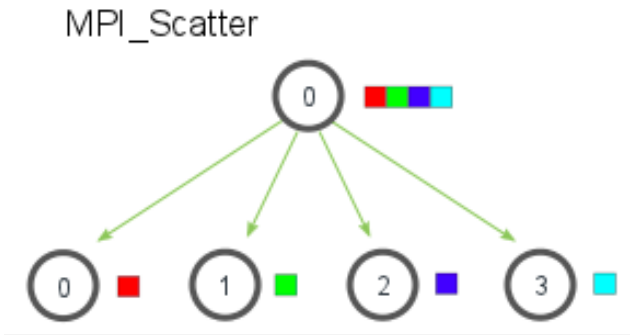


Performance Optimizations I

$$a_i = F(x_i)$$
$$v_{i+1} = v_i + a_i \, \Delta t$$

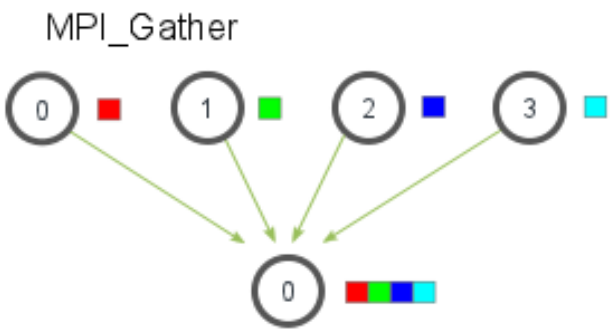
Parallelization I

```
while (True) {
    quadtree = QuadTree(bodies);

    #pragma omp parallel for shared(bodies)
    for (size_t i = 0; i < bodies.size(); i++) {
        auto& body = sbodies[i];
        quadtree.calculate_force(body);
    }
```



Parallelization II

```
        #pragma omp parallel for shared(bodies)
        for (size_t i = 0; i < sbodies.size(); i++) {
            auto& body = sbodies[i];

            if (step % 2 == LEAP) {
                body.leap(timestep);
            }
            else {
                body.frog(timestep);
            }
        }
    }
```

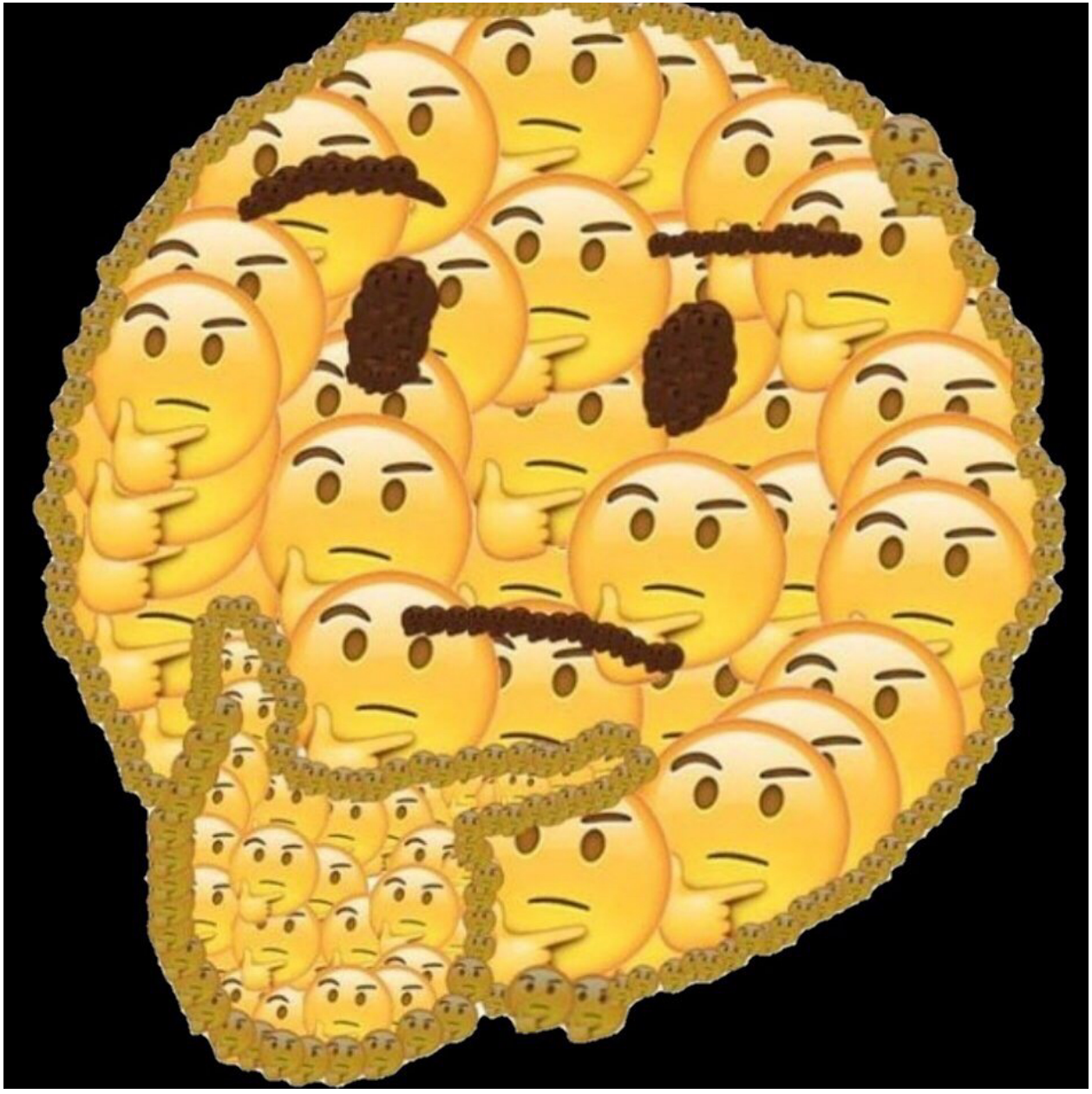


Parallelization III

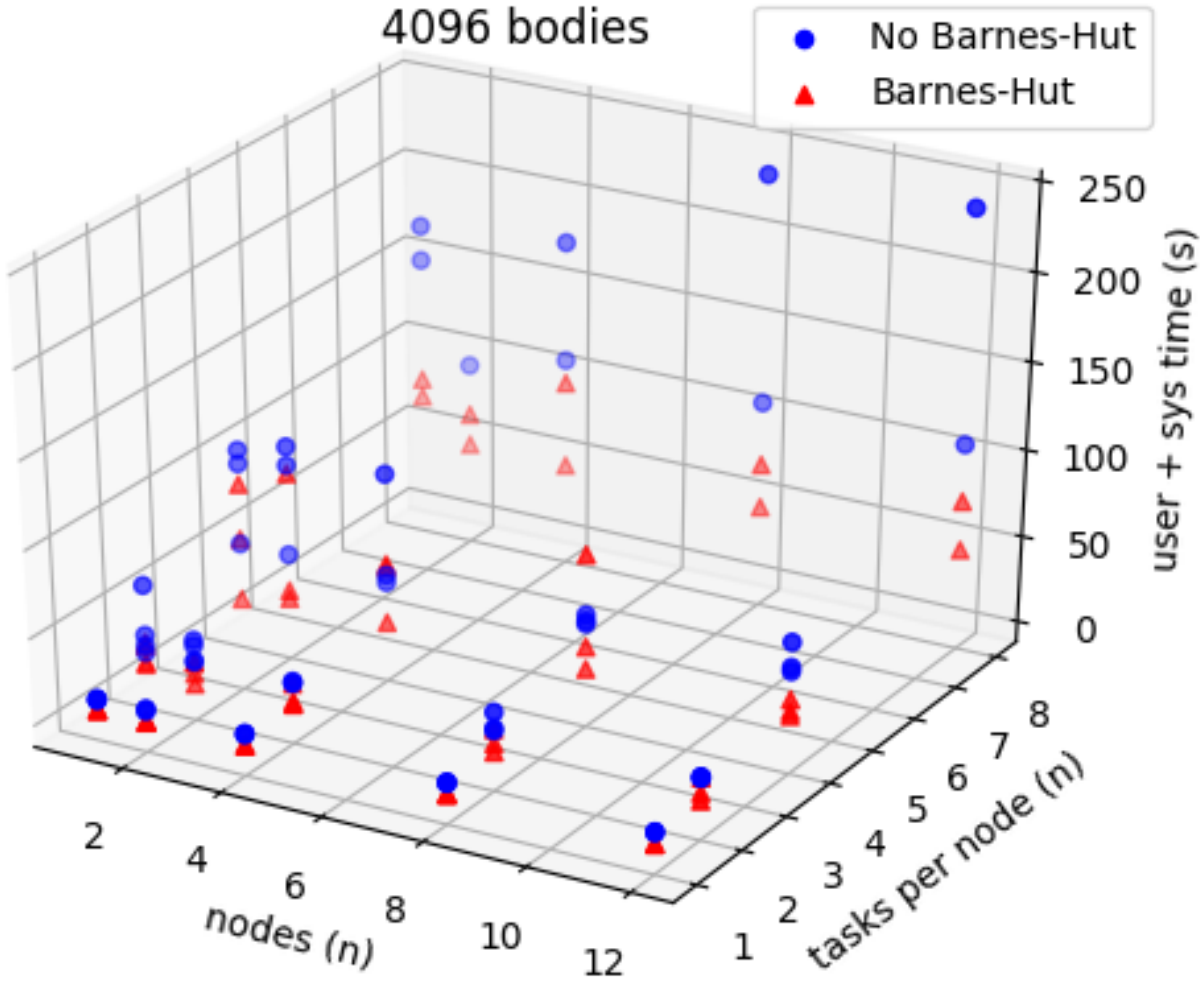
Total CPU time on rank 0 was 89.820000
Total CPU time on rank 1 was 37.040000
Total CPU time on rank 2 was 36.750000
Total CPU time on rank 3 was 36.960000

Resultss I

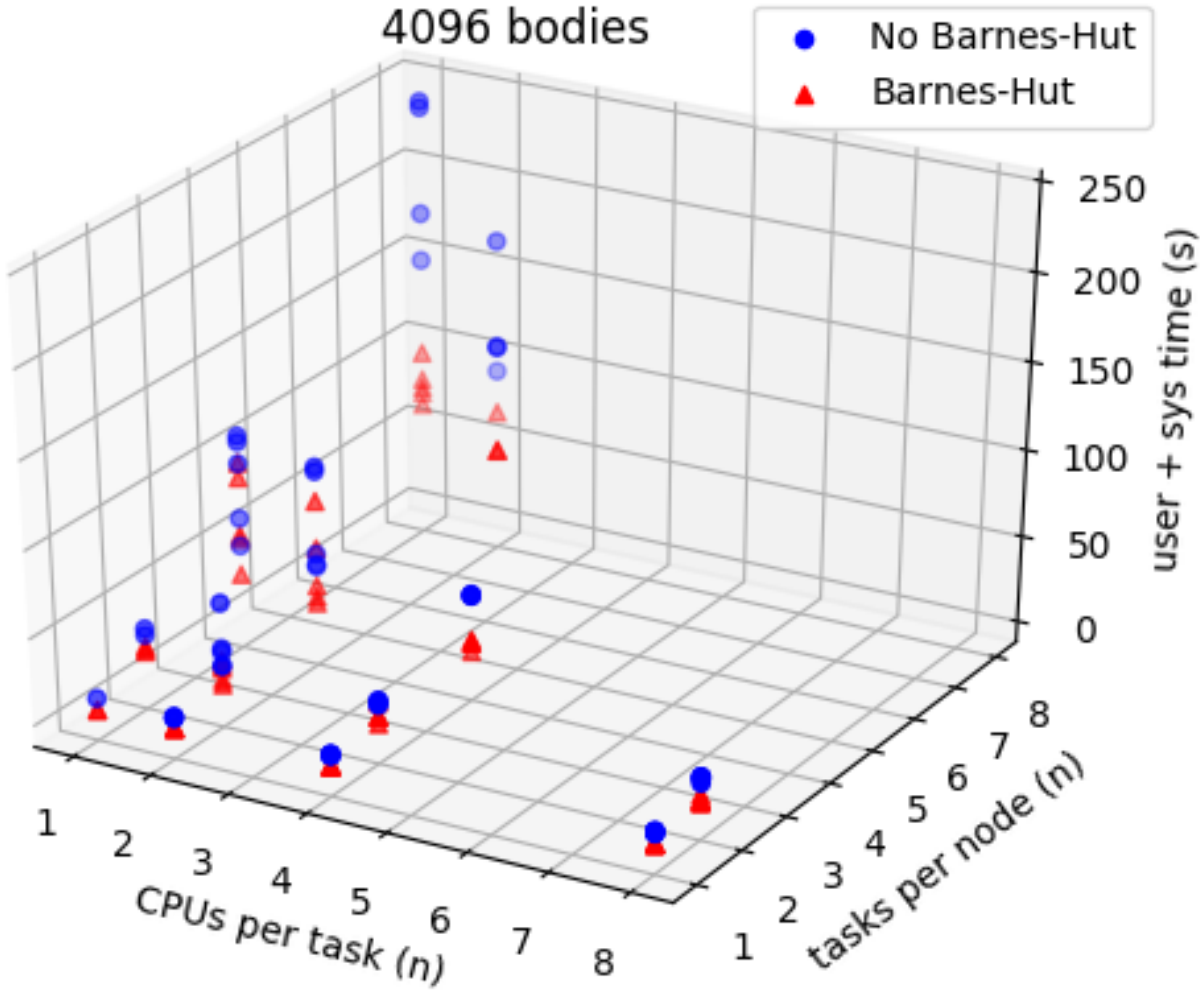
Results I



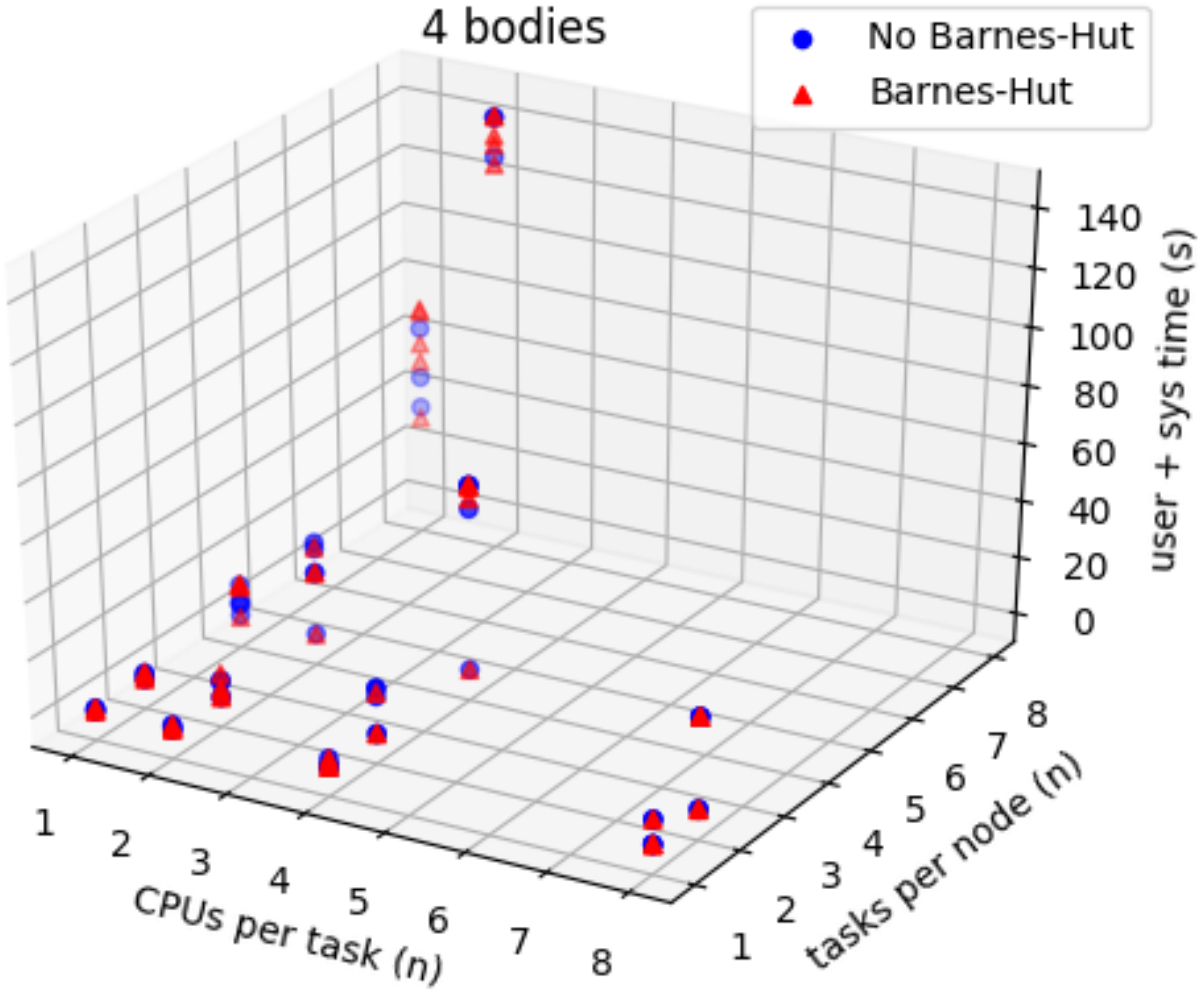
Results II


4096 bodies
No Barnes-Hut
Barnes-Hut
user + sys time (s)
0
50
100
150
200
250
nodes (n)
2
4
6
8
10
12
tasks per node (n)
1
2
3
4
5
6
7
8

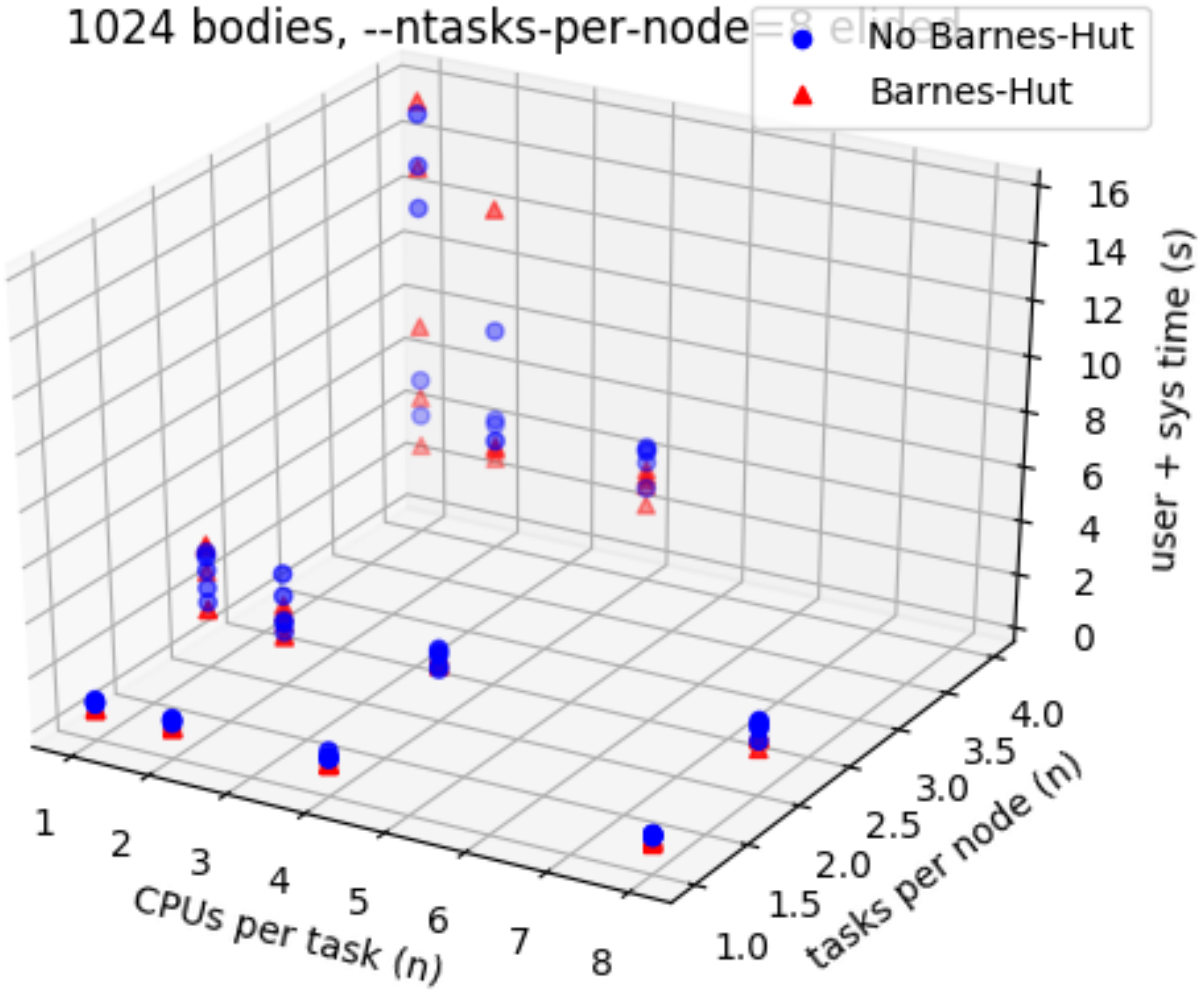
Results III


4096 bodies
No Barnes-Hut
Barnes-Hut
250
200
150
100
50
0
user + sys time (s)
1
2
3
4
5
6
7
8
CPUs per task (n)
1
2
3
4
5
6
7
8
tasks per node (n)

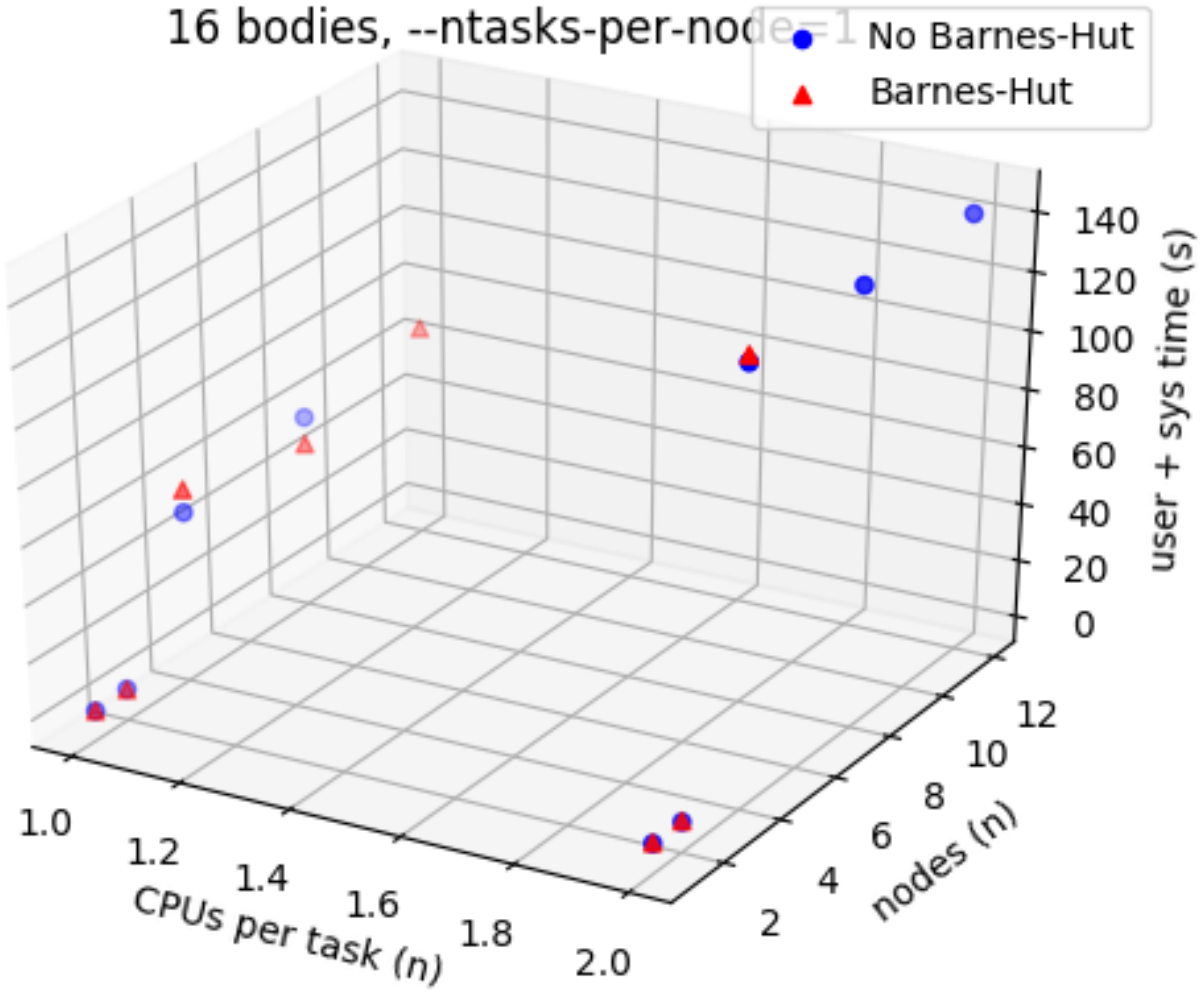
Results IV


4 bodies
No Barnes-Hut
Barnes-Hut
user + sys time (s)
0
20
40
60
80
100
120
140
CPUs per task (n)
1
2
3
4
5
6
7
8
tasks per node (n)
1
2
3
4
5
6
7
8

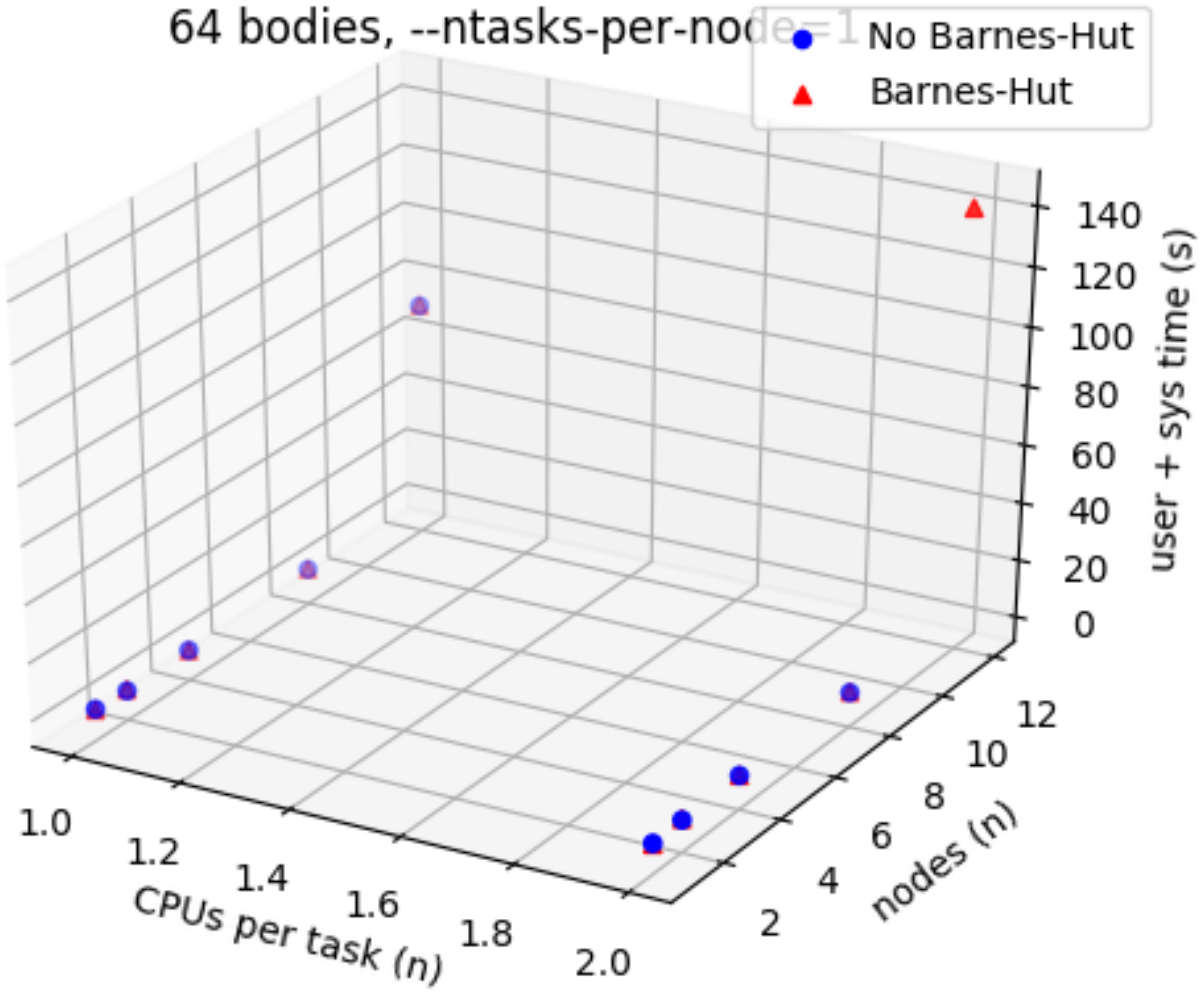
Results V


1024 bodies, --ntasks-per-node=8 elided
No Barnes-Hut
Barnes-Hut
user + sys time (s)
0
2
4
6
8
10
12
14
16
CPUs per task (n)
1
2
3
4
5
6
7
8
tasks per node (n)
1.0
1.5
2.0
2.5
3.0
3.5
4.0

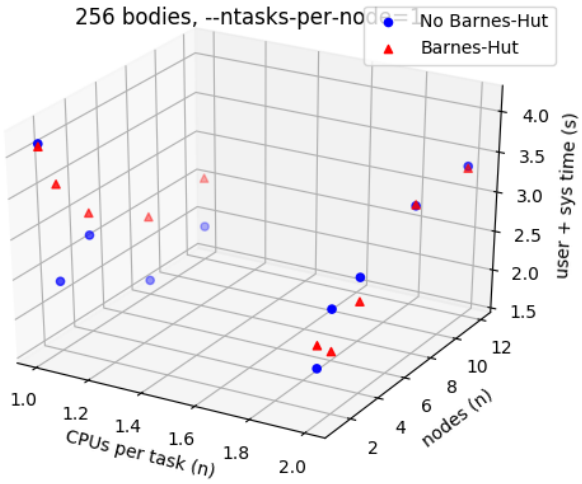
Results VI

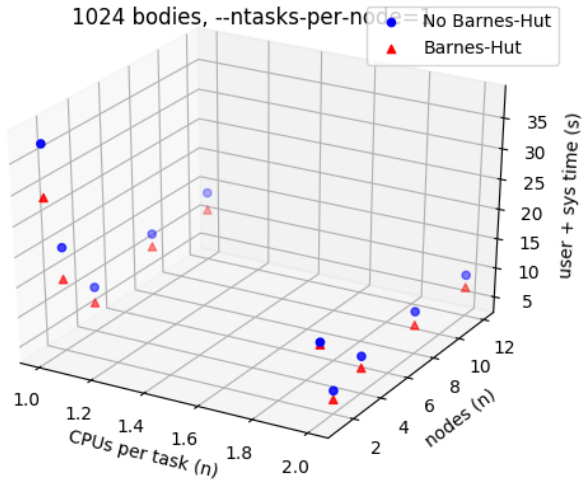

16 bodies, --ntasks-per-node=1
No Barnes-Hut
Barnes-Hut
user + sys time (s)
0
20
40
60
80
100
120
140
nodes (n)
2
4
6
8
10
12
CPUs per task (n)
1.0
1.2
1.4
1.6
1.8
2.0

Results VII

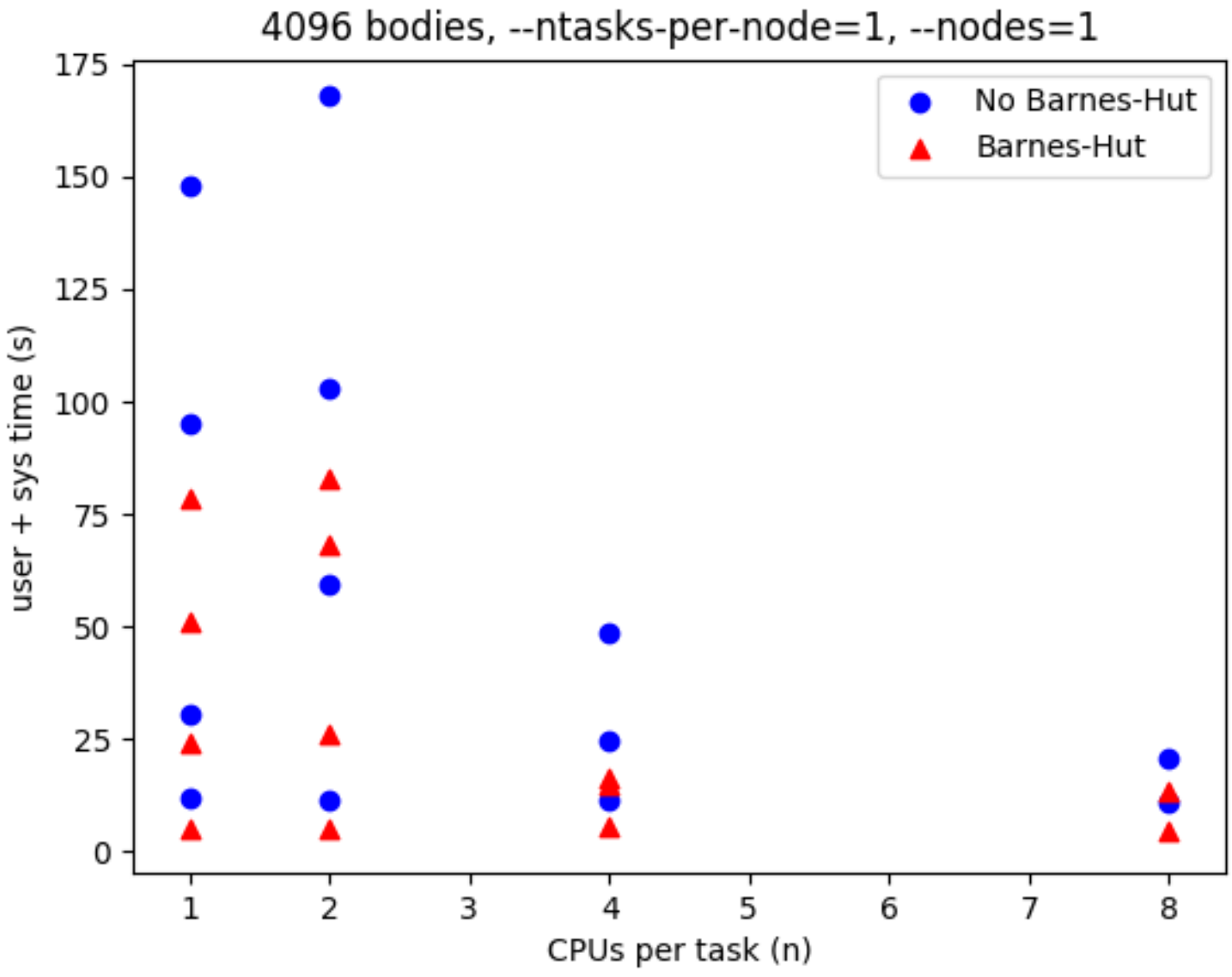

64 bodies, --ntasks-per-node=1
No Barnes-Hut
Barnes-Hut
user + sys time (s)
0
20
40
60
80
100
120
140
nodes (n)
2
4
6
8
10
12
CPUs per task (n)
1.0
1.2
1.4
1.6
1.8
2.0

Results VIII





Results IX


4096 bodies, --ntasks-per-node=1, --nodes=1
No Barnes-Hut
Barnes-Hut
user + sys time (s)
CPUs per task (n)
0
25
50
75
100
125
150
175
1
2
3
4
5
6
7
8