Ughur Mammadzada

Simulating multi-star systems using Python (and maybe C++)

Supervisor: Martin Paleico

Table of contents

1 Multi-star simulations

2 Sequential Execution

3 Parallel Execution

4 Parallel Performance and Scaling

5 Conclusion and Future Plan

Outline

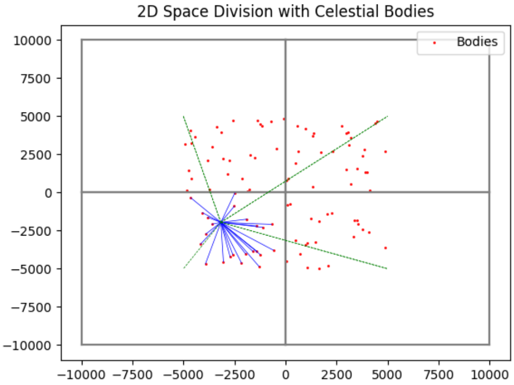
1 Multi-star simulations
2 Sequential Execution
3 Parallel Execution
4 Parallel Performance and Scaling
5 Conclusion and Future Plan

Why hard?

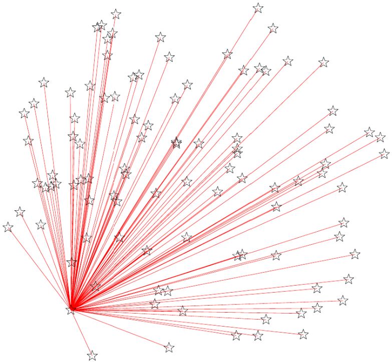
- Simulation is a model of the real world
- Simulation is always approximate, but must give insights
- Simulating real system on a classical computer takes too much time

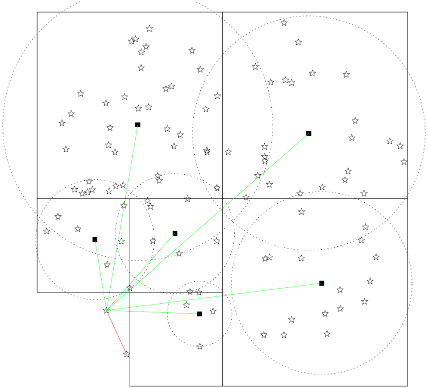
Approaches

- Body-body interactions
  - Complexity $O(n^2)$
  - 1M bodies - 1T calculations
- OctTree based object distribution
  - Complexoty $O(n \log(n))$
  - 1M bodies - 6M calculations
- Space chunking - Current choice
  - Compelxity $O(n\sqrt{n})$
  - 1M bodies - 2B calculations



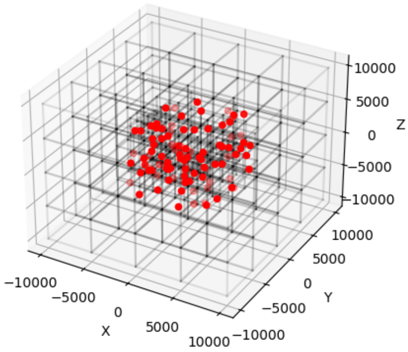
Body-body vs Tree

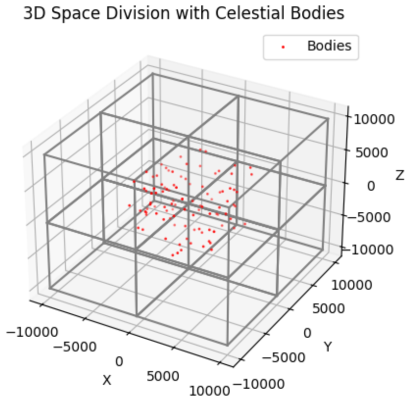




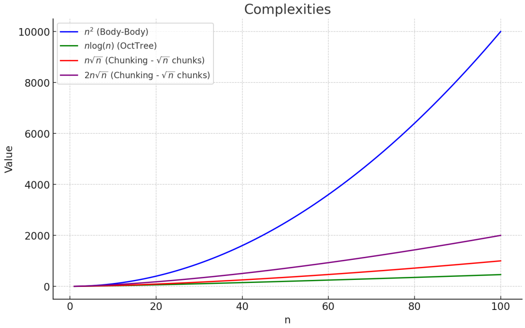
Dehnen and Read, "N-body simulations of gravitational dynamics"

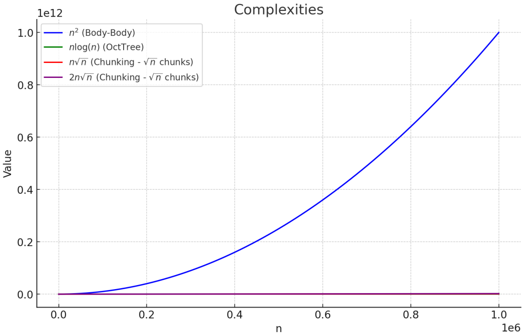
OctTree vs Space Chunking





Complexities





Outline

1 Multi-star simulations

2 Sequential Execution

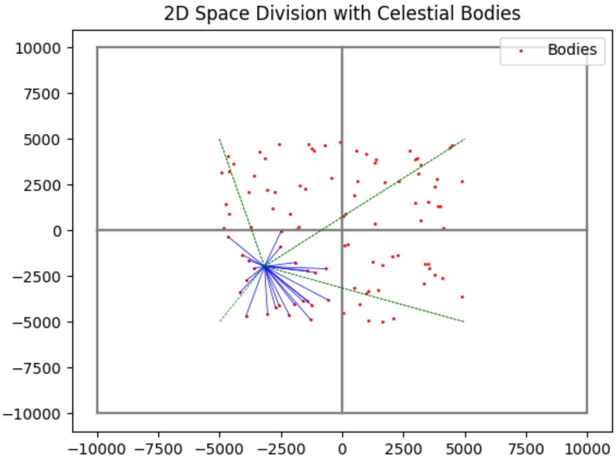
3 Parallel Execution

4 Parallel Performance and Scaling

5 Conclusion and Future Plan

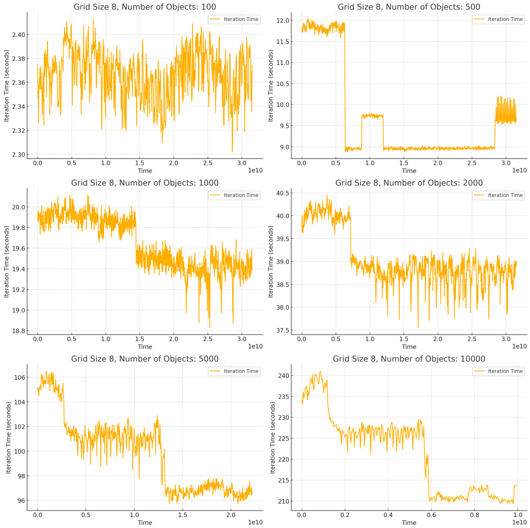
Calculations

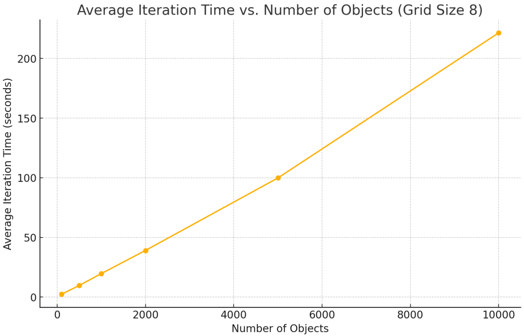
- Same calculation, as in case of an OctTree
- Leapfrog Integrator [1]:
  - $v_{i+0.5} = v_i + a_i * \frac{dt}{2}$
  - $x_i + 1 = x_i + v_{i+0.5} * dt$
  - $v_{i+1} = v_{i+0.5} + a_{i+0.5} * \frac{dt}{2}$



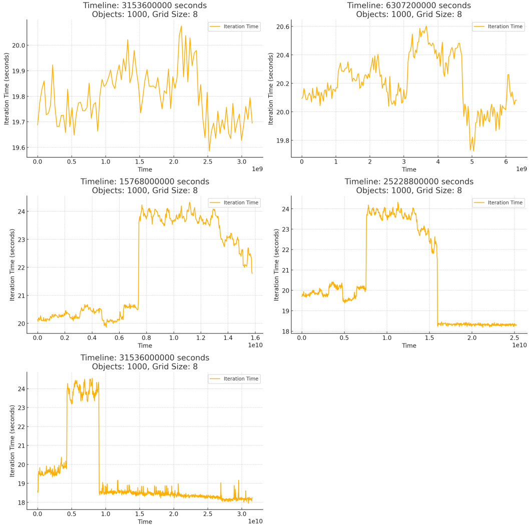
Wikipedia, *Leapfrog integration — Wikipedia, The Free Encyclopedia*

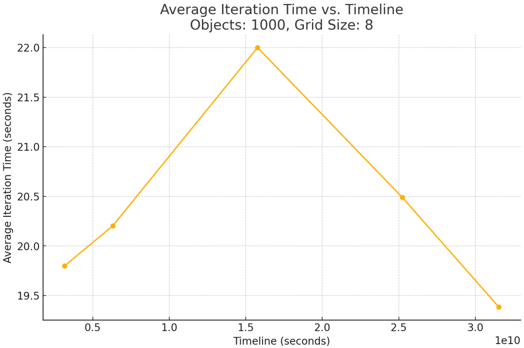
Performance



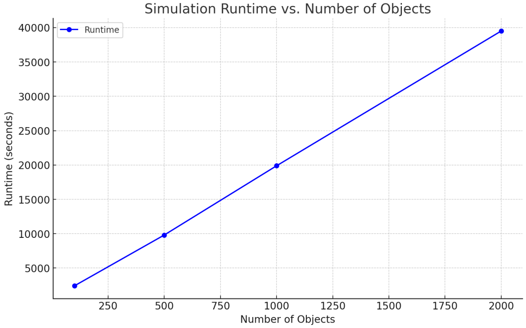


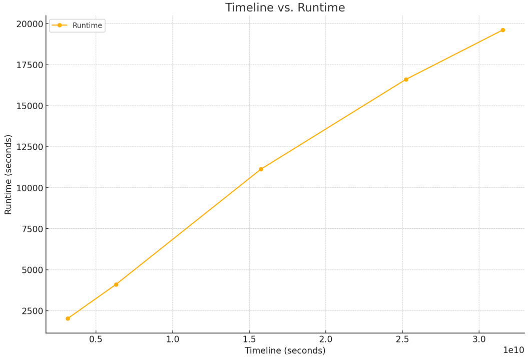
Performance



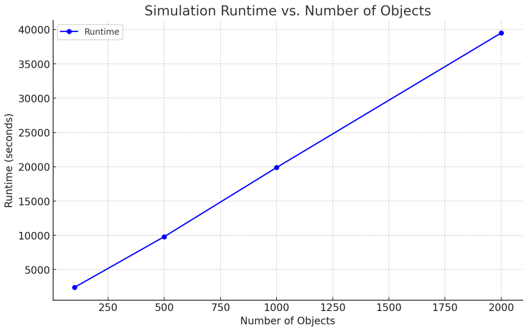


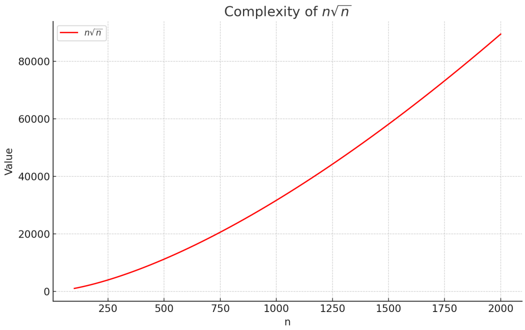
Performance





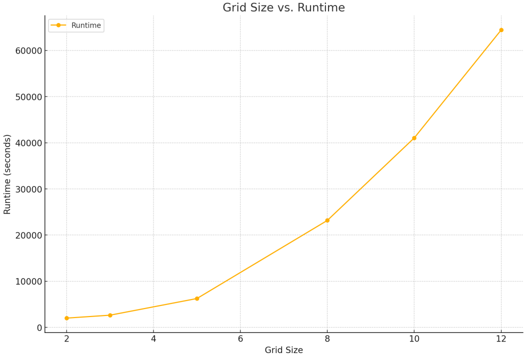
Performance





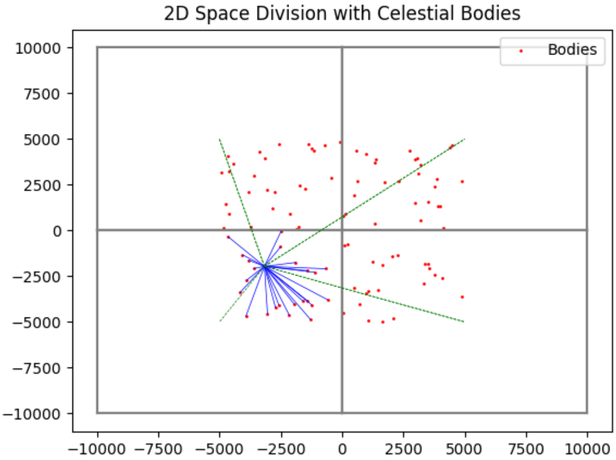
Performance

- $m = \sqrt{n}$
- $O(\frac{n^2}{m} + nm)$
- $O(n\sqrt{n} + n\sqrt{n})$



Performance

- $O(\frac{n^2}{m} + nm)$
- $m = n => O(n + n^2)$
- $m = 1 => O(n^2 + n)$
- $\frac{\sqrt{n}}{3} <= m <= 3\sqrt{n}$



Outline

1 Multi-star simulations

2 Sequential Execution

3 Parallel Execution

4 Parallel Performance and Scaling

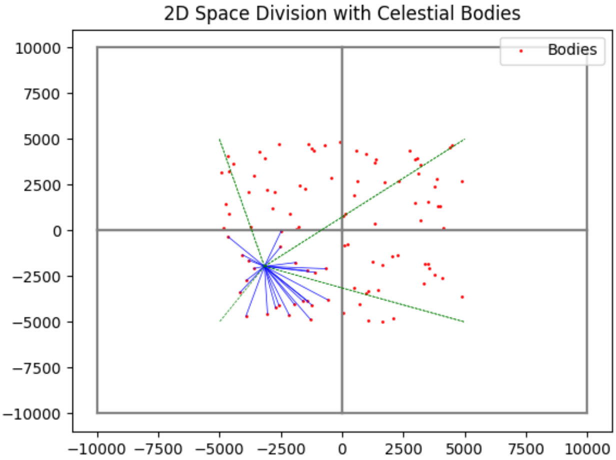
5 Conclusion and Future Plan

Can be Parallel

- Generation / Input read
- Calculation itself - Done
- Object-chunk tracking - Memory intensive
- Output write
- Animation

Workload Distribution

- Chunks can be distributed
- Only total masses and centres are communicated



Outline

1 Multi-star simulations

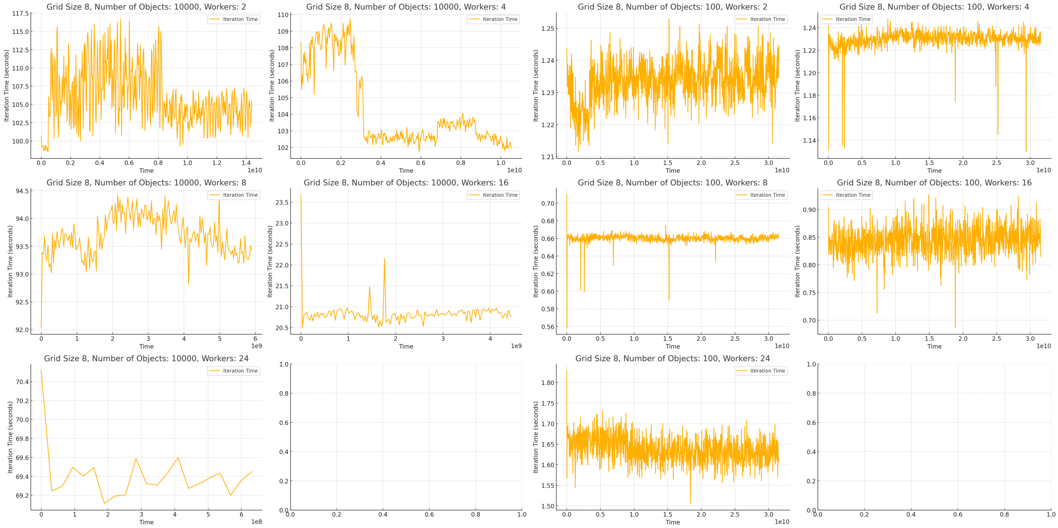
2 Sequential Execution

3 Parallel Execution

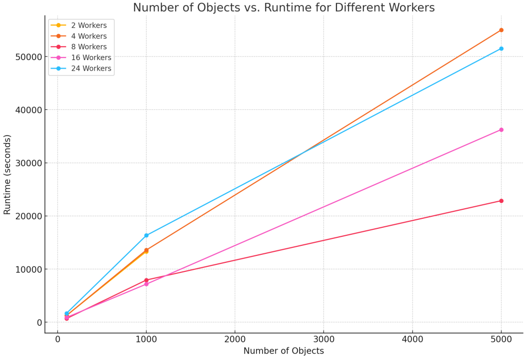
4 Parallel Performance and Scaling

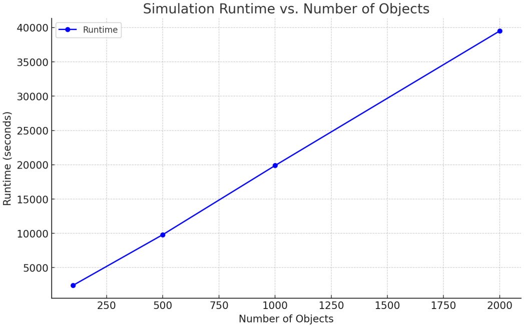
5 Conclusion and Future Plan

Performance

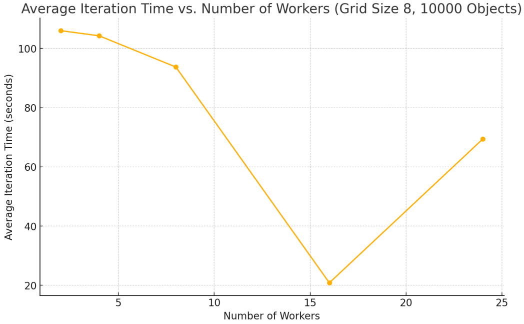


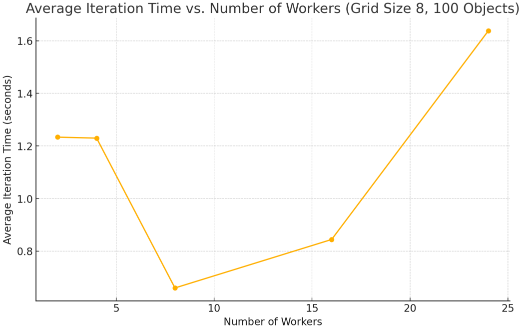
Comparison to Sequential



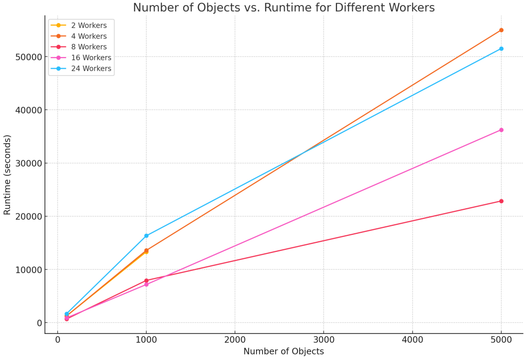


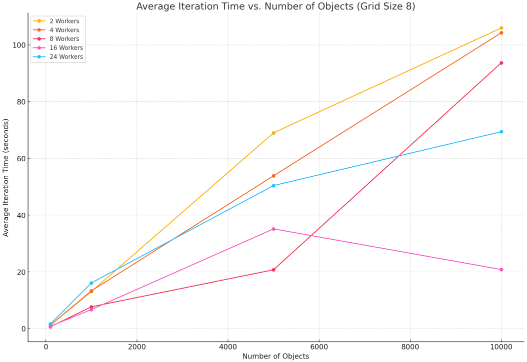
Scaling


Average Iteration Time vs. Number of Workers (Grid Size 8, 10000 Objects)
Average Iteration Time (seconds)
Number of Workers
20
40
60
80
100
5
10
15
20
25


Average Iteration Time vs. Number of Workers (Grid Size 8, 100 Objects)
Average Iteration Time (seconds)
Number of Workers
0.8
1.0
1.2
1.4
1.6
5
10
15
20
25

Scaling


Number of Objects vs. Runtime for Different Workers
2 Workers
4 Workers
8 Workers
16 Workers
24 Workers
Runtime (seconds)
Number of Objects
0
10000
20000
30000
40000
50000
1000
2000
3000
4000
5000


Average Iteration Time vs. Number of Objects (Grid Size 8)
2 Workers
4 Workers
8 Workers
16 Workers
24 Workers
Average Iteration Time (seconds)
Number of Objects
0
20
40
60
80
100
2000
4000
6000
8000
10000

LIKWID Analysis

Shell

```
attempt to index a nil value (field '?')
stack traceback:
        .../likwid-mpirun:1952: in function 'printMpiOutput'
        .../likwid-mpirun:2528: in main chunk
        [C]: in ?
```

Outline

1 Multi-star simulations

2 Sequential Execution

3 Parallel Execution

4 Parallel Performance and Scaling

5 Conclusion and Future Plan

Overall Results

- Simulation scales overall good, but can be better
- Optimal number of nodes depends on number of objects and chunks (claim)
- Optimal number of chunks depend on the number of objects
- Real-world scale simulation might not be the best choice

What is Next

- Checkpoints
- Further scaling analysis
- OctTree parallelization
- Overhead measurement and reduction
- I/O parallelization
- Visualization parallelization
- On node parallelization with multiprocessing library
- Error handling

References

Dehnen, W. and J. I. Read. "N-body simulations of gravitational dynamics". In: *The European Physical Journal Plus* 126.5 (May 2011). ISSN: 2190-5444. DOI: 10.1140/epjp/i2011-11055-3. URL: http://dx.doi.org/10.1140/epjp/i2011-11055-3.

Wikipedia. *Leapfrog integration — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Leapfrog%20integration&oldid=1220184662. [Online; accessed 21-June-2024]. 2024.