Project Advanced Programming

N-body Simulation

Thomas Van Bogaert

Basis van N-body simulatie

- N lichamen met elk een positie en massa
- Doel: bereken de evolutie van elk lichaam over tijd
- Moeilijkheid: elk lichaam kan elk ander lichaam beïnvloeden

Berekenen van de evolutie van elk lichaam

- Wiskundig voor te stellen door differentiaalvergelijking
- Benader de oplossing met numerieke integratie

Velocity Verlet integratie

- Tweede orde benadering
- Naast positie nu ook snelheid en acceleratie nodig
- Als Δt kleiner $\Rightarrow$ benadering preciezer

Op tijd t worden de nieuwe positie en snelheid gegeven door:

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(t)\Delta t^2$$

$$\vec{v}(t+\Delta t) = \vec{v}(t) + \frac{\vec{a}(t) + \vec{a}(t+\Delta t)}{2}\Delta t$$

Velocity Verlet integratie

Probleem: wat als $\vec{a}(t_0)$ niet gegeven?

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(t)\Delta t^2$$

$$\vec{v}(t+\Delta t) = \begin{cases} \vec{v}(t) + \vec{a}(t+\Delta t)\Delta t & \text{als } t = t_0 \\ \vec{v}(t) + \frac{\vec{a}(t)+\vec{a}(t+\Delta t)}{2}\Delta t & \text{als } t \neq t_0 \end{cases}$$

Berekenen van $\vec{a}(t)$

- In het algemeen: bereken de kracht tussen elk paar objecten
- Brute force: $O(n^2)$
- Barnes-hut benadering: $O(n \log(n))$

Brute force

```
for bodyA in bodies:
   newAcceleration = (0, 0, 0)
   for bodyB in bodies:
      forceAB = force(bodyA, bodyB)
      newAcceleration += forceAB / massA
   updateVelocity(bodyA, newAcceleration)
   updateAcceleration(bodyA, newAcceleration)
```

- `forceAB` is een pure functie
- Zeer goed paralleliseerbaar

Implementatie in C++

In de implementatie berekent `forceAB` de aantrekkingskracht onder invloed van zwaartekracht.
Brute force N-body simulatie implementaties:

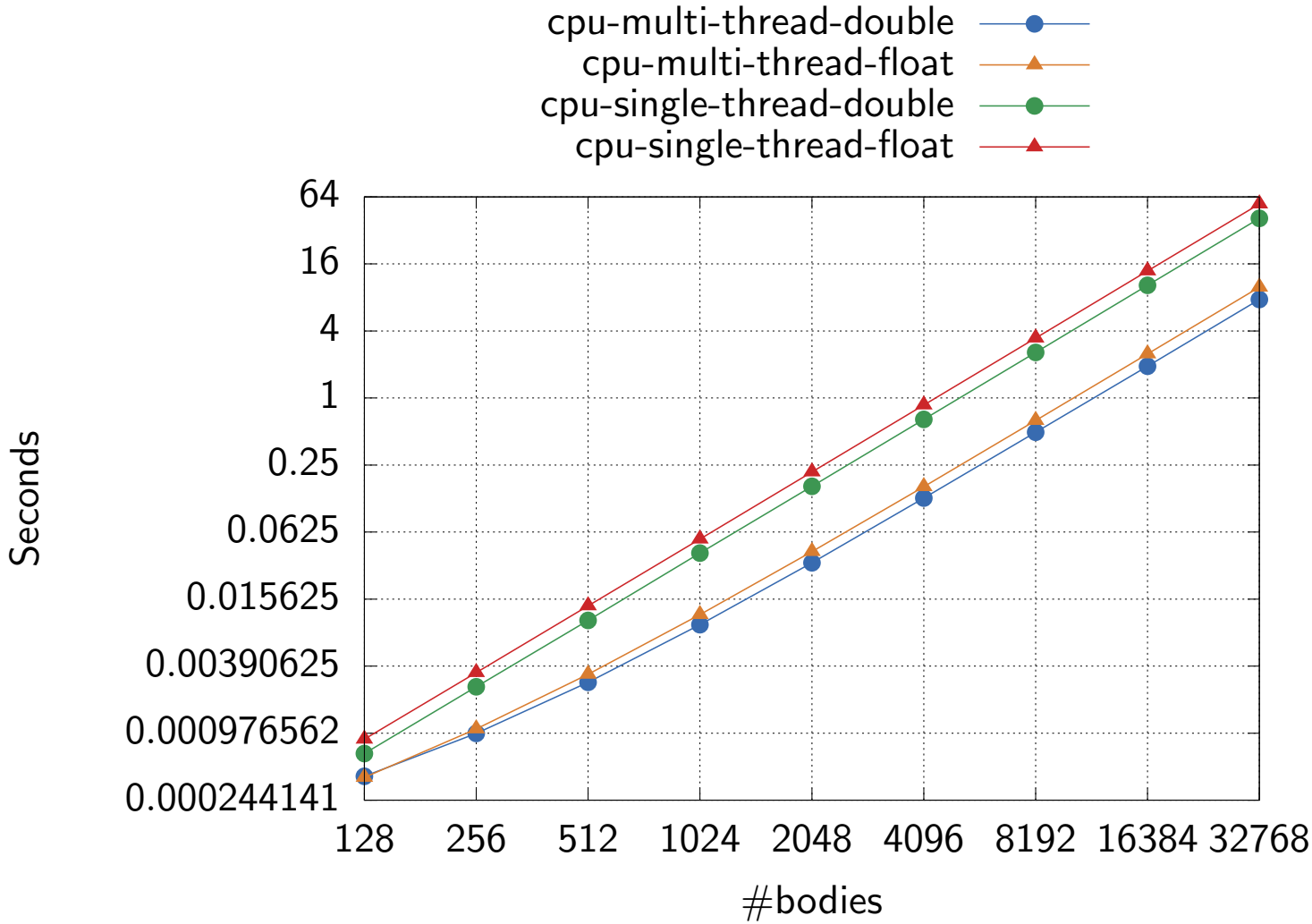
- Single threaded (cpu-single-thread)
- Multi threaded met OpenMP (cpu-multi-thread)
- Naïve OpenCL implementatie (opencl)
- OpenCL implementatie gebruik makend van lokaal geheugen (openclloc)
- OpenCL implementatie met Struct of Arrays (SoA) i.p.v. Array of Structs (AoS) implementatie (openclvec)

Iedere implementatie is beschikbaar met double of single floating point precisie.

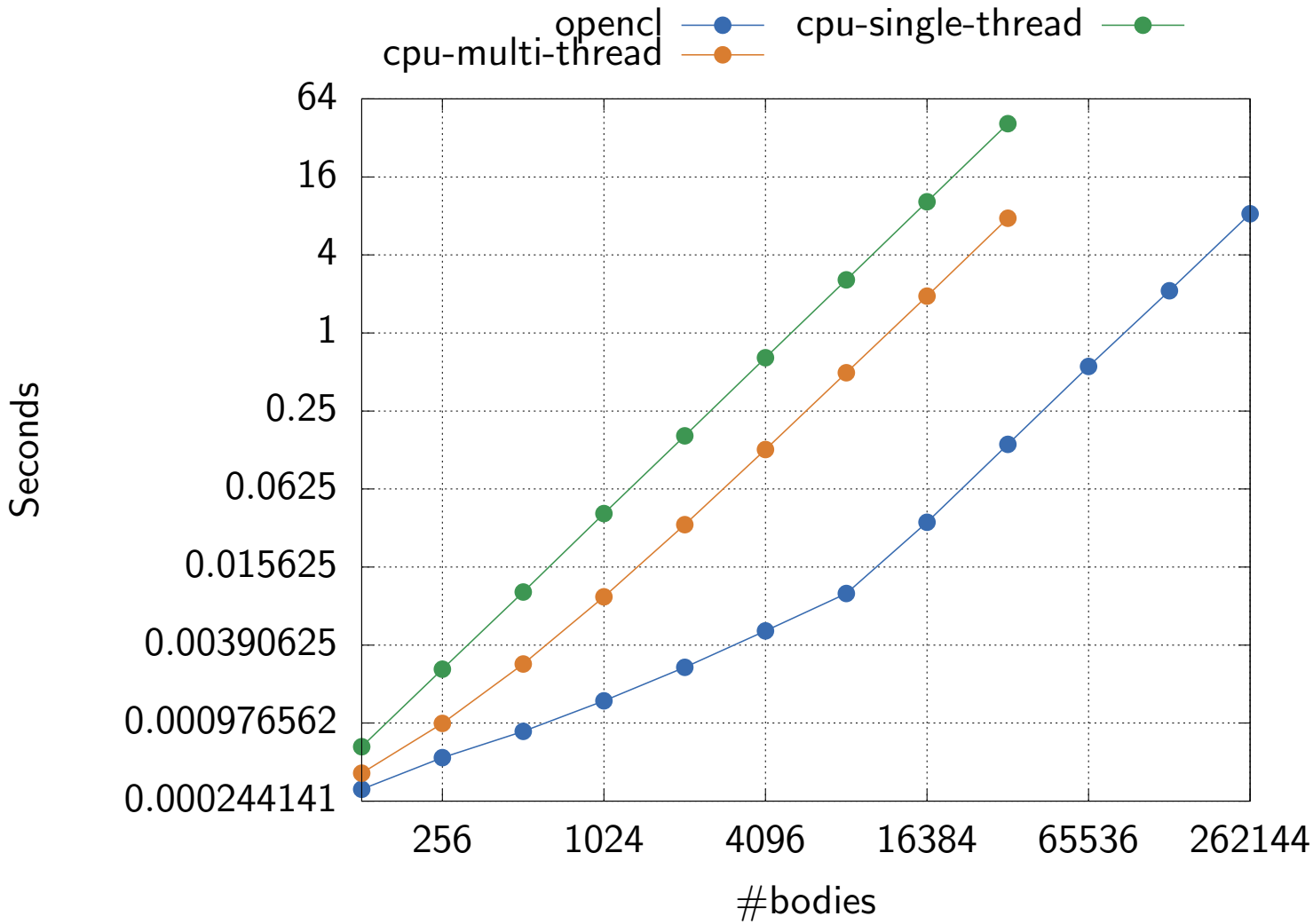
Vergelijking van performantie

- Nu komt de vergelijking van alle brute force implementaties
- Alle assen hebben logaritmische schaal (log(2))
- Gebruikte CPU: Xeon E3 1650 @3.6 Ghz
- Gebruikte GPU (OpenCL): AMD RX580 @1200 Mhz
- Linux kernel 4.20.3 (Antergos default scheduler)
- cpu-multi-thread implementatie is gebenchmarked met 6 cores op een 6 core machine

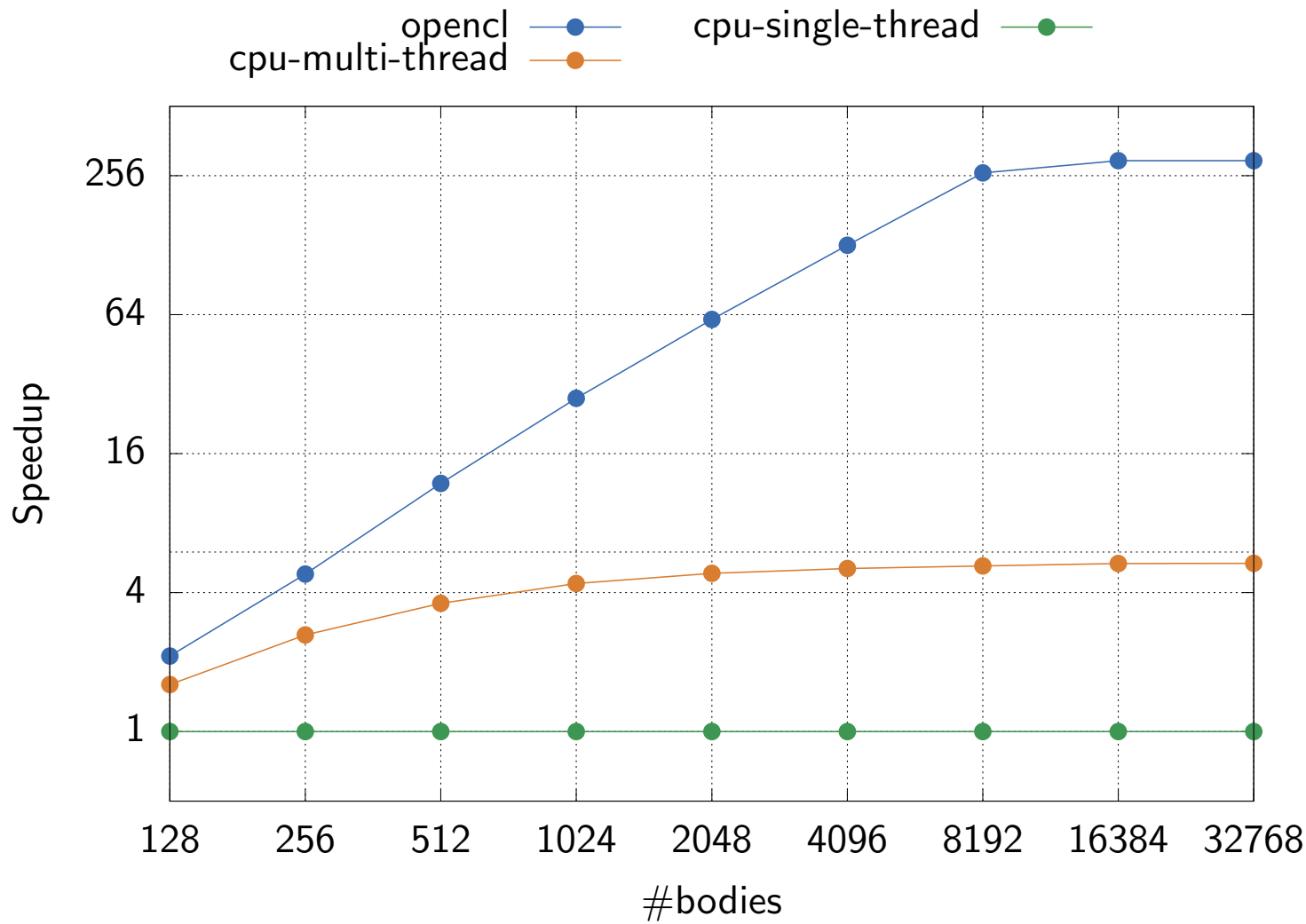
Comparison of CPU implementations



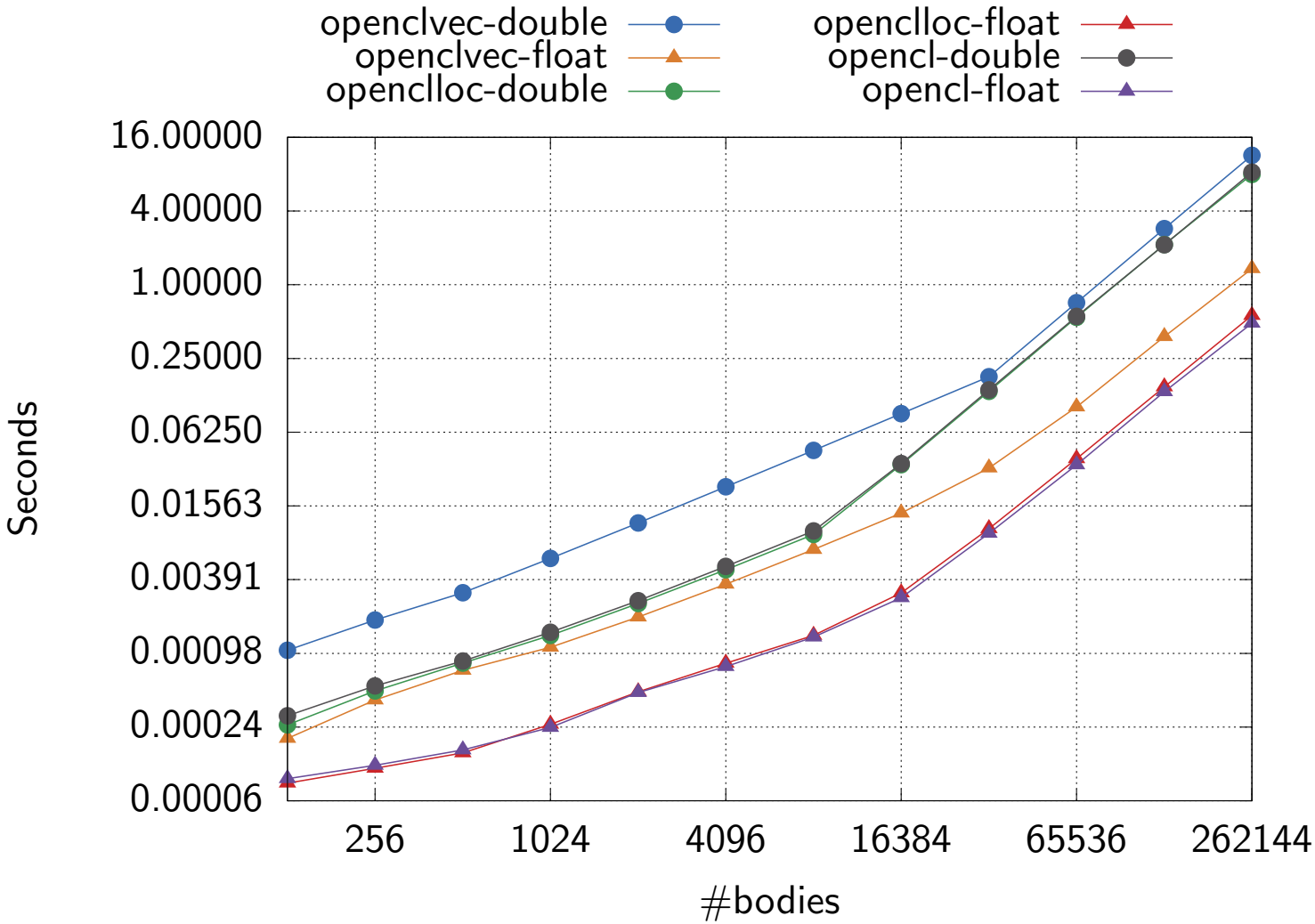
Compare different implementations (double)



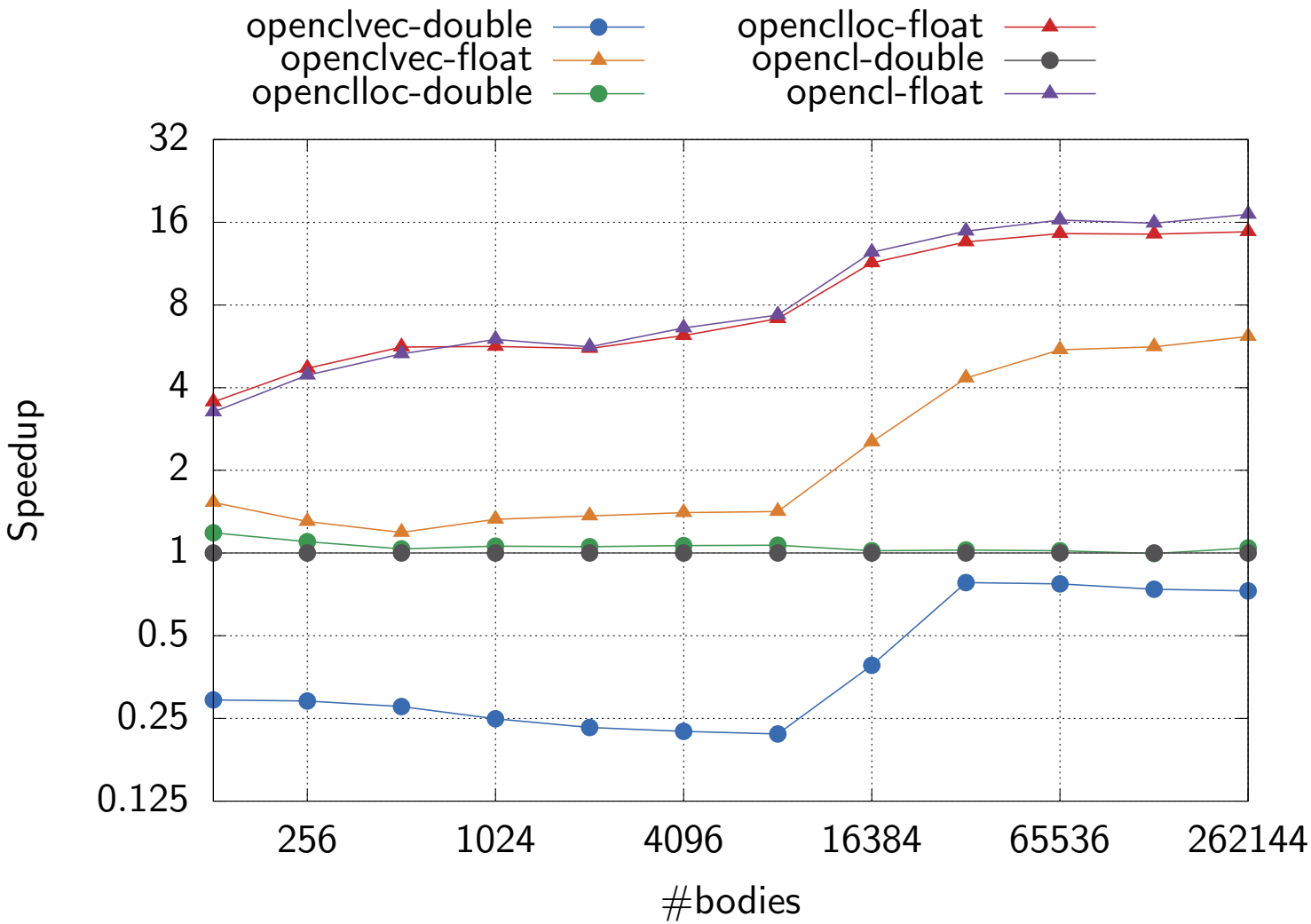
Speedup between different implementations (double)



Comparison of OpenCL implementations



Speedup between OpenCL implementations



Resultaat van simulatie