Exploring the Use of Molecular Dynamics Simulations for High-Performance Space Debris Collision Modelling

Pablo Gómez[1], Fabio Gratl[2]
[1] Advanced Concepts Team, TEC-SF, European Space Agency
[2] Chair for Scientific Computing in Computer Science, Technical University Munich

Ariadna ID 21-5105
Contract Number: 4000134654/21/NL/AS

Background & Motivation

Exploring the Use of Molecular Dynamics Simulations for High-Performance Space Debris Collision Modelling

Pablo Gómez[1], Fabio Gratl[2]

[1] Advanced Concepts Team, TEC-SF, European Space Agency

[2] Chair for Scientific Computing in Computer Science, Technical University Munich

Ariadna ID 21-5105

Contract Number: 4000134654/21/NL/AS

High-Performance Computing & Space Debris Simulations

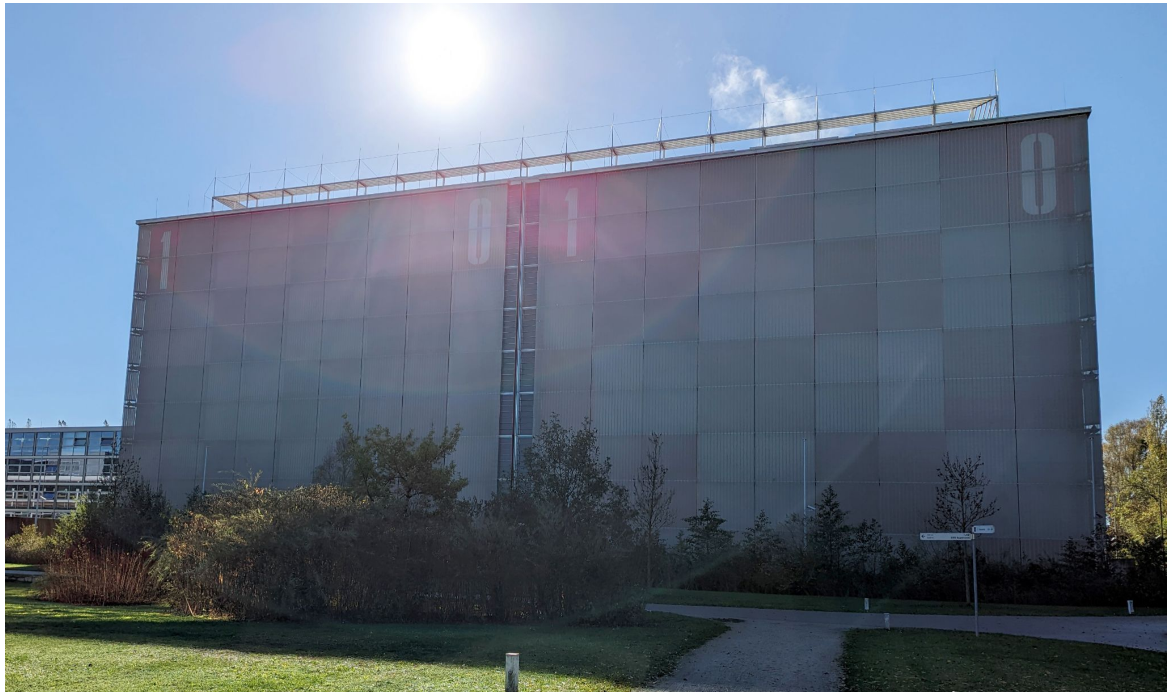
esa

Scientific Motivation

- Limited previous efforts in large-scale computational approaches to long-term debris simulations
- Feasibility of deterministic modeling has not been explored on new hardware generation

Need for ESA-external Partners

- Lack of supercomputing facilities at ESA
- Interesting prior research in particle simulations from TUM

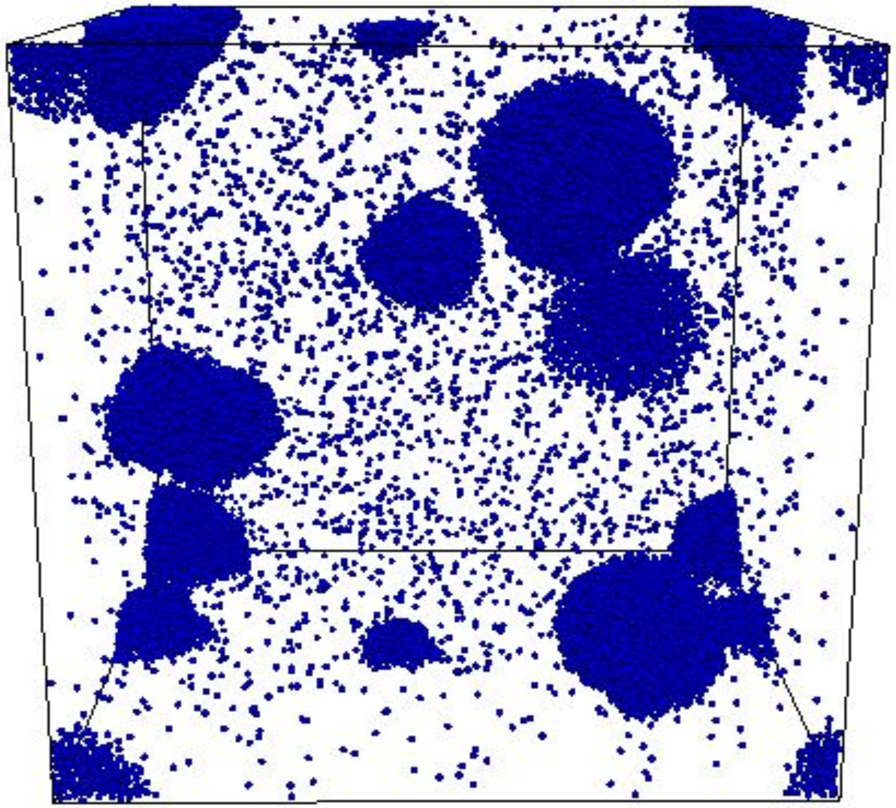


Leibniz Supercomputing Centre (LRZ)

Prior Research at TUM

AutoPas

- Highly-parallel, auto-tuned particle container for N-body simulations in C++17
- Primarily applied to molecular dynamics simulations
- Efficient for short-range particle interactions



Gratl *et al.* 2019

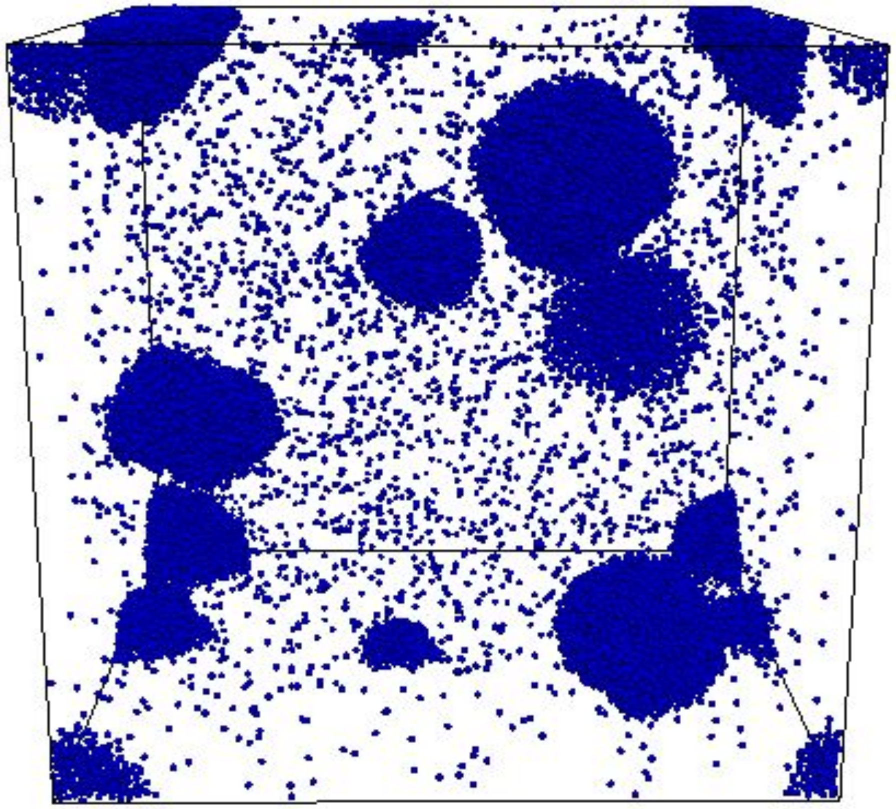
Prior Research at TUM

AutoPas

- Highly-parallel, auto-tuned particle container for N-body simulations in C++17
- Primarily applied to molecular dynamics simulations
- Efficient for short-range particle interactions

Space Debris

- Utilize AutoPas to enable efficient scaling over large number of particles and large number of cores
- Model collisions / conjunctions as short-range interaction



Gratl *et al.* 2019

Executive Summary

Exploring the Use of Molecular Dynamics Simulations for High-Performance Space Debris Collision Modelling

Pablo Gómez[1], Fabio Gratl[2]

[1] Advanced Concepts Team, TEC-SF, European Space Agency

[2] Chair for Scientific Computing in Computer Science, Technical University Munich

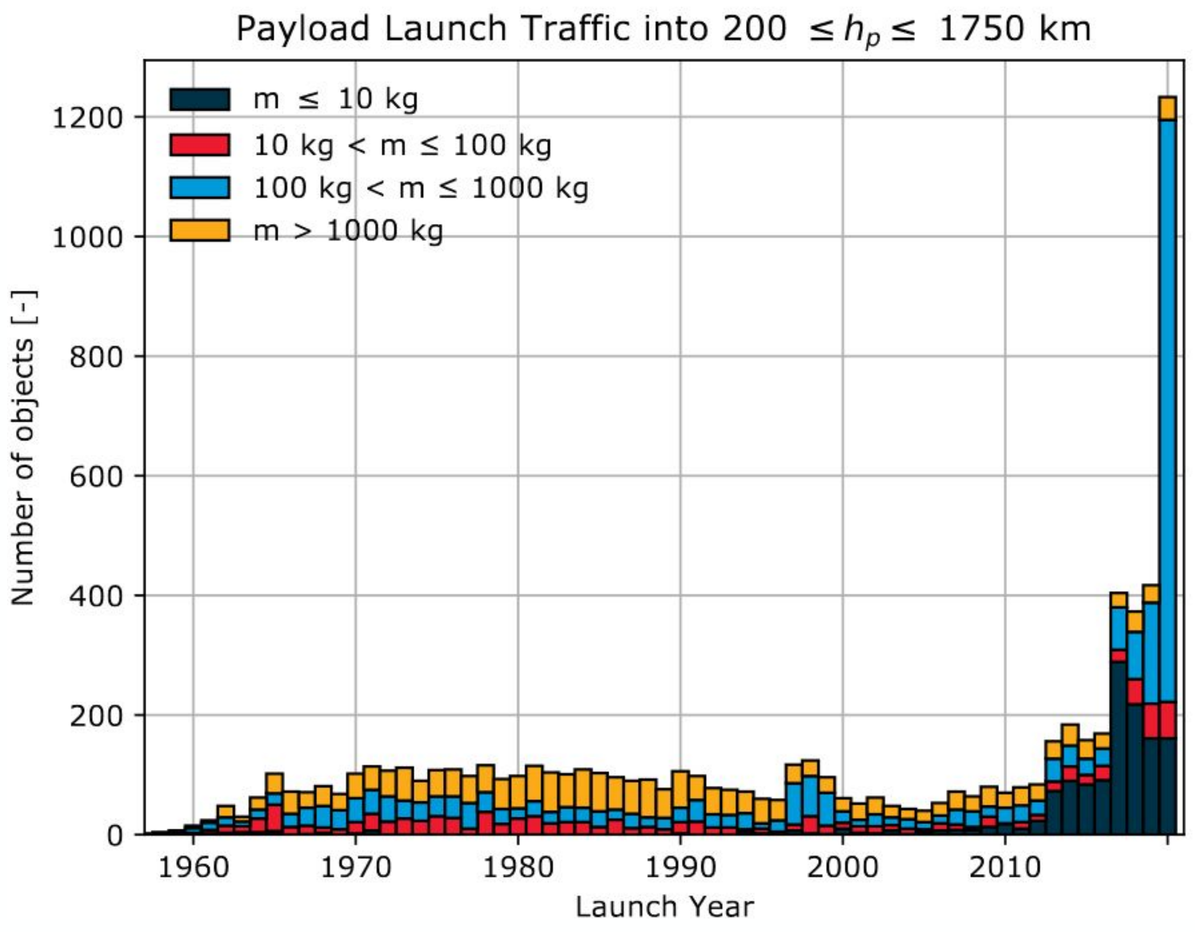
Ariadna ID 21-5105

Contract Number: 4000134654/21/NL/AS

Background

Challenges of long-term, small debris simulations

- Fine-grained temporal and spatial resolution
- Simulating 10^4 - 10^5 particles is computationally costly
- Long-term debris modeling relies on stochastic models



ESA's Annual Space Environment Report 2022

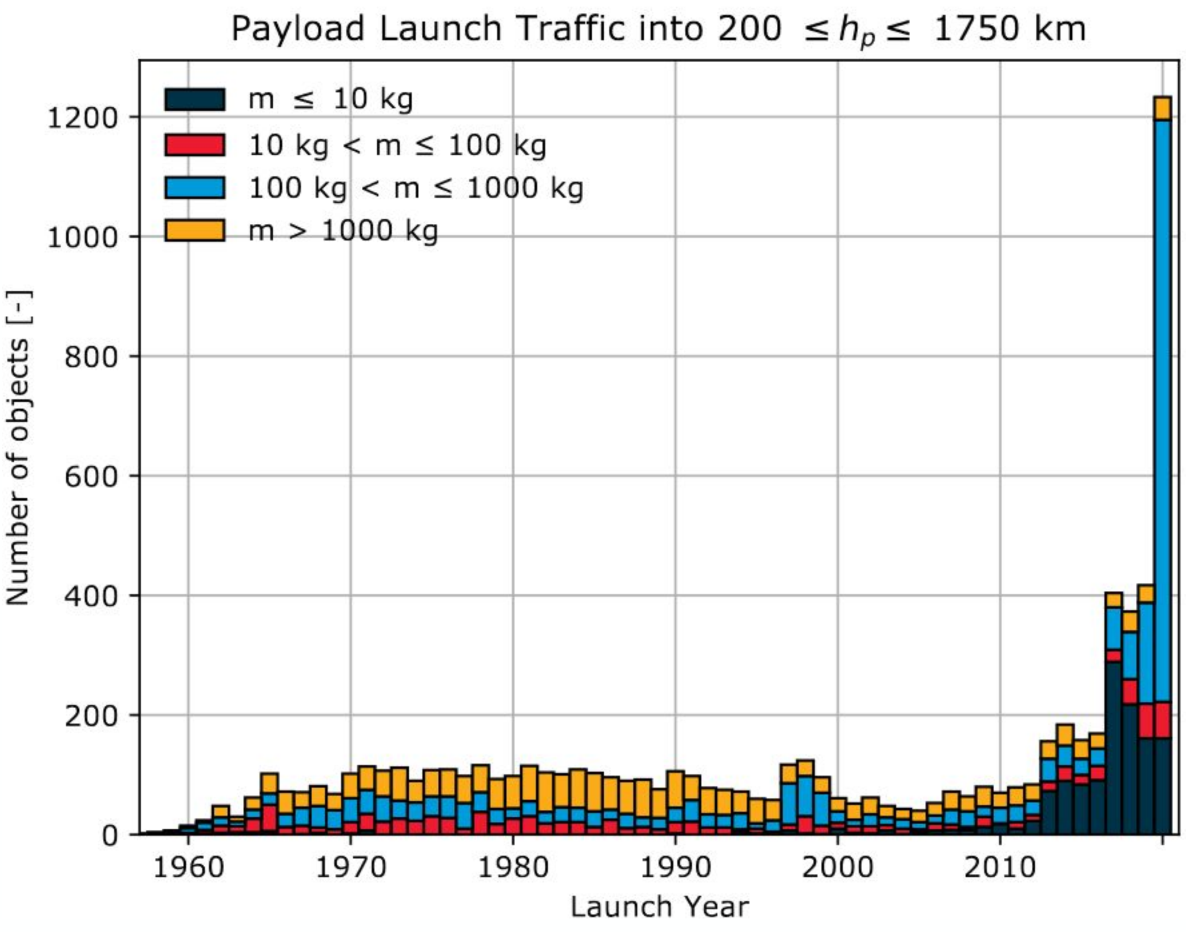
Background

Challenges of long-term, small debris simulations

- Fine-grained temporal and spatial resolution
- Simulating 10^4 - 10^5 particles is computationally costly
- Long-term debris modeling relies on stochastic models

Ariadna Study between ACT, Space Debris Office and Technical University Munich to

- Build a scalable non-stochastic simulation
- Explore recent tools from particle dynamics
- Rely on modern C++ and HPC via OpenMP & MPI



ESA's Annual Space Environment Report 2022

Study Objectives

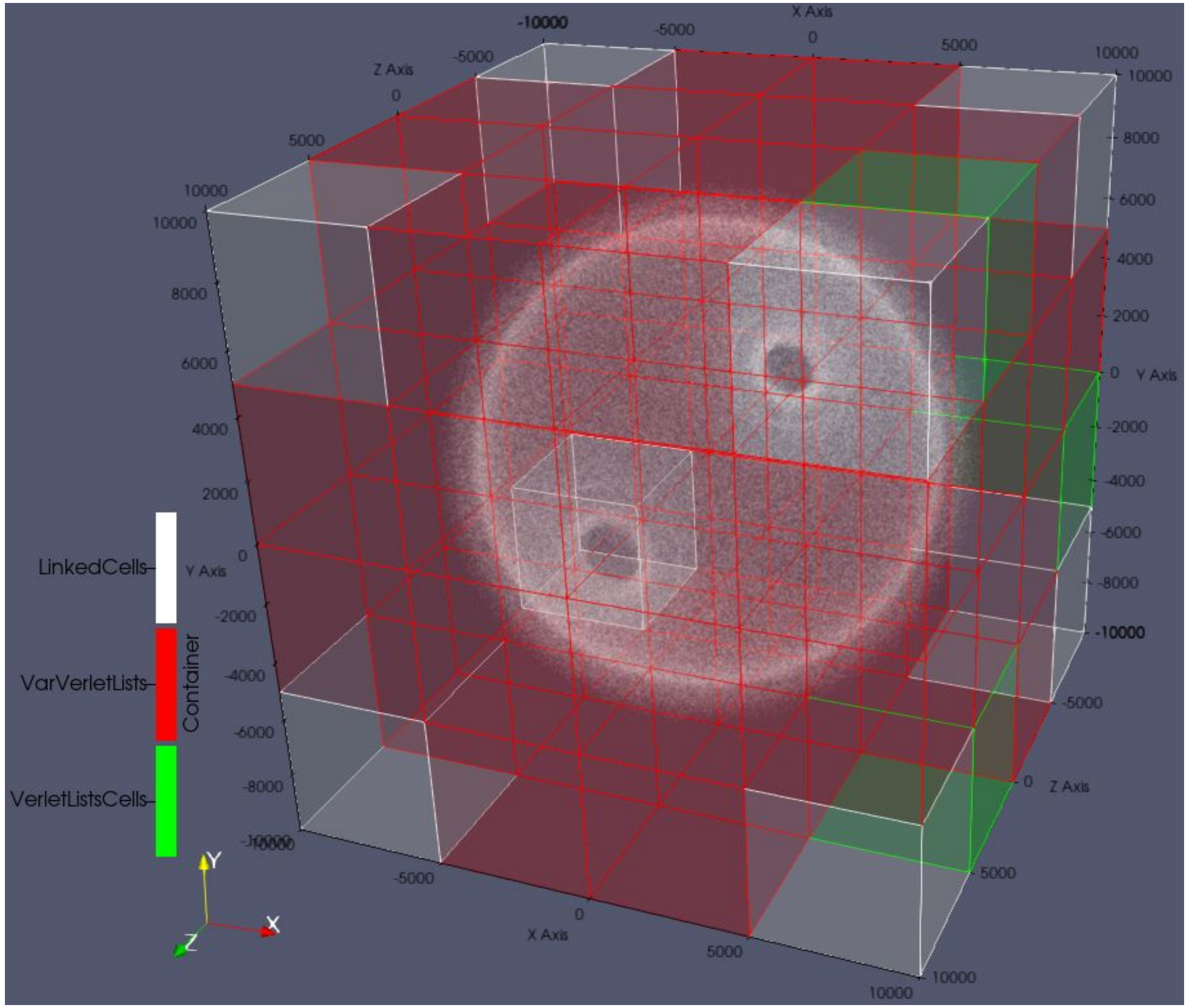
esa

- Investigate **feasibility** of deterministic conjunction models for long-term simulations

- Investigate **scaling** potential using advances from high-performance computing and large-scale particle simulations

- Ideally, enable simulations with **small debris** (> 1cm)

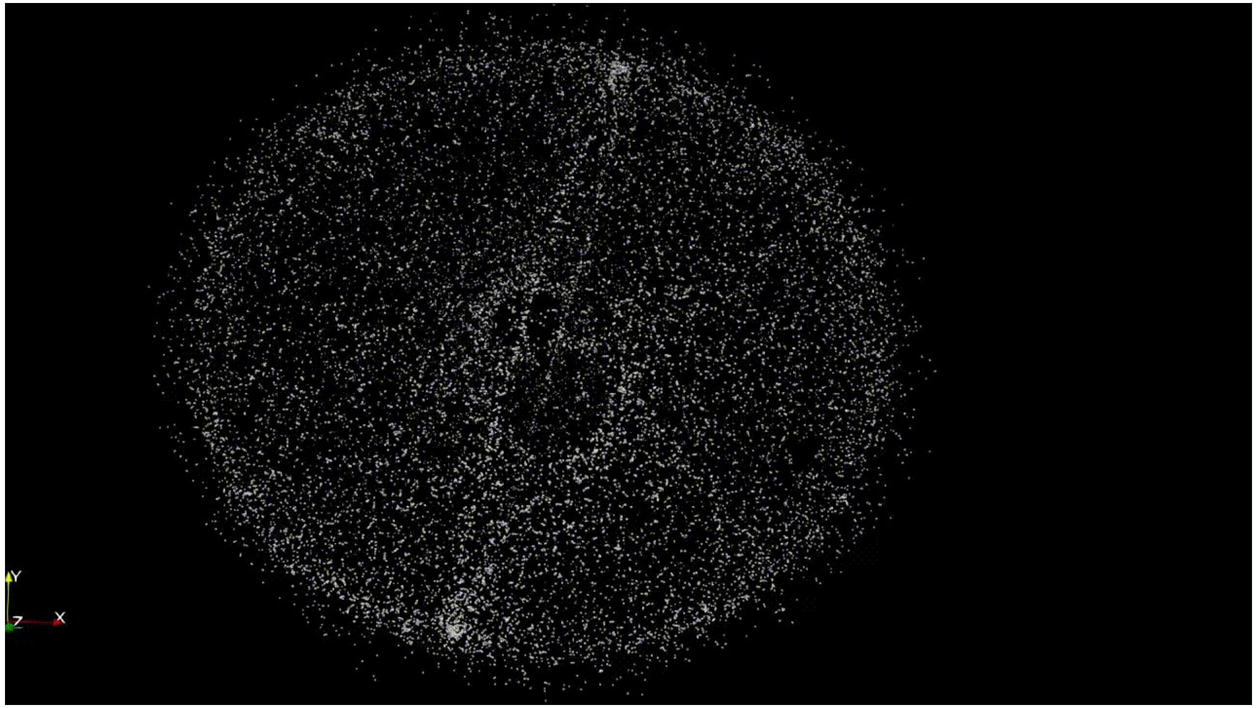
- Build tools for future use by following **open-source** best practices



Automatic Data Structure Selection

Core Findings

- Simulating the **small debris** environment over long time spans in a deterministic simulation **is feasible**
- For 600k particles: **50x real** time on just 128 compute nodes
- Smart domain decomposition is key (e.g. based on concentric spheres)
- Sub-timestep resolution is central to conjunction detection
- Highly parallel and precise propagator is required



Visualization of a Simulation

Outcomes - Software

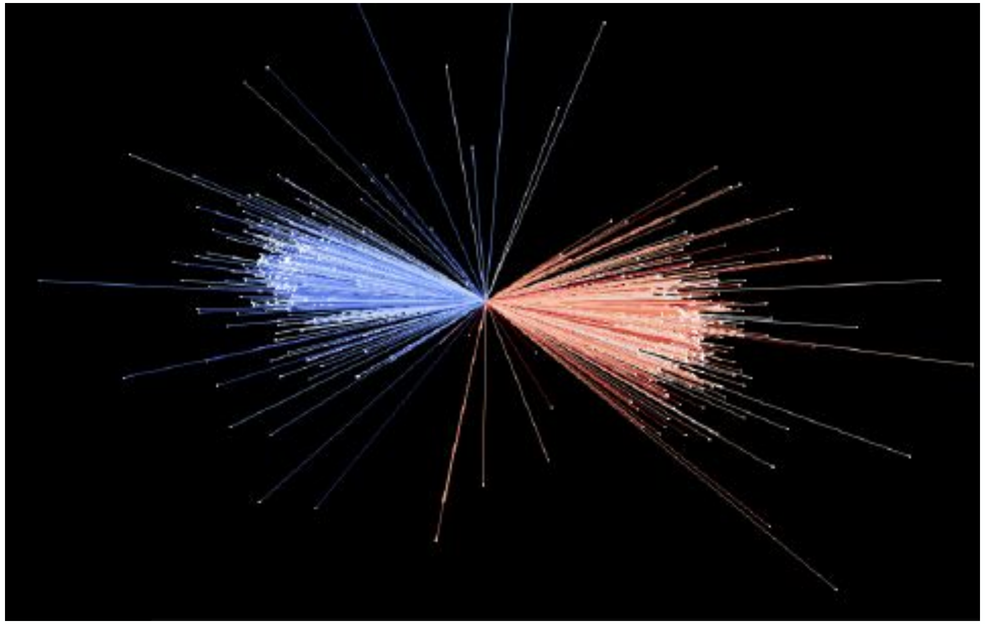
LArge-scale Deterministic Debris Simulation (LADDS)

- Modular regarding dynamics, breakup models, …
- Scalable up to thousands of cores
- Open-source C++17 software

LADDS Logo

NASA Breakup Model

- Validated against original and other implementations
- Flexible API enabled easy integration into LADDS
- Open-source C++17 software


Breakup Simulation with LADDS

All study objectives achieved!

Outcomes - Publications & Theses

- *Deterministic Conjunction Tracking in Long-term Space Debris Simulations*
  Pablo Gómez, Fabio Gratl, Oliver Bösing, Dario Izzo
  3rd IAA Conference on Space Situational Awareness (ICSSA) 2022, Madrid, Spain

- *Modeling Upcoming Megaconstellations in Space Debris Environment Simulations*
  Albert Noswitz
  Bachelor’s Thesis, TUM

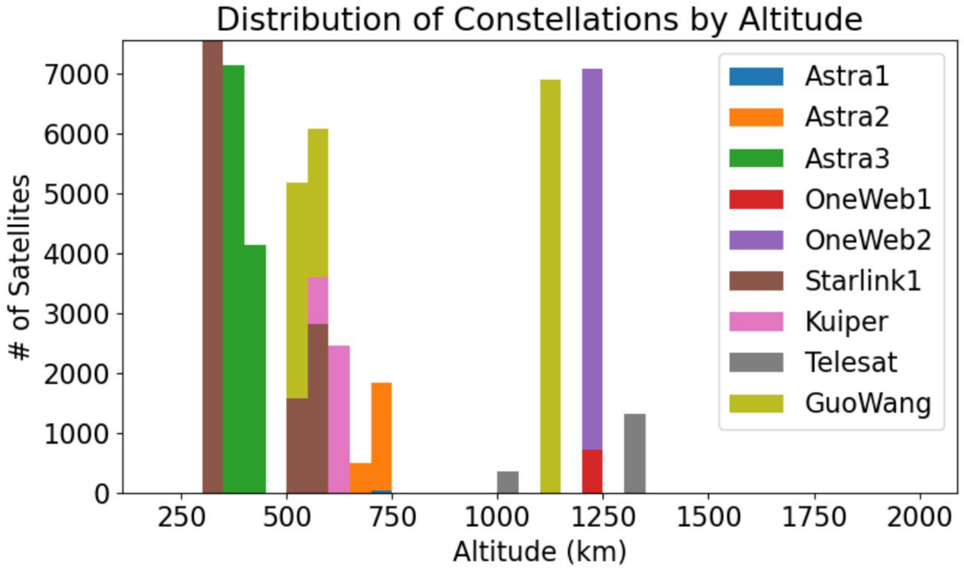
- *Efficient Trajectory Modelling for Space Debris Evolution*
  Oliver Bösing
  Bachelor’s Thesis, TUM

- *Efficient Implementation and Evaluation of the NASA Breakup Model in modern C++*
  Jonas Schuhmacher
  Bachelor’s Thesis, TUM

- *Massively Parallel Long-term Small Space Debris Simulations*
  Pablo Gómez, Fabio Gratl, Albert Noswitz, Jonas Schuhmacher, Oliver Bösing, Dario Izzo
  Work in Progress

Outlook - Future Research & Applications

- Direct comparison of different models (e.g. cubes / breakup models)
- Comparison of different datasets
- Integrate high-fidelity propagator (Heyoka)
- Investigate impact of constellations
- Long-term evolution of projected CDMs / CAMs



Constellations by Altitude, Noswitz 2022

Technical Presentation

Exploring the Use of Molecular Dynamics Simulations for High-Performance Space Debris Collision Modelling

Pablo Gómez[1], Fabio Gratl[2]

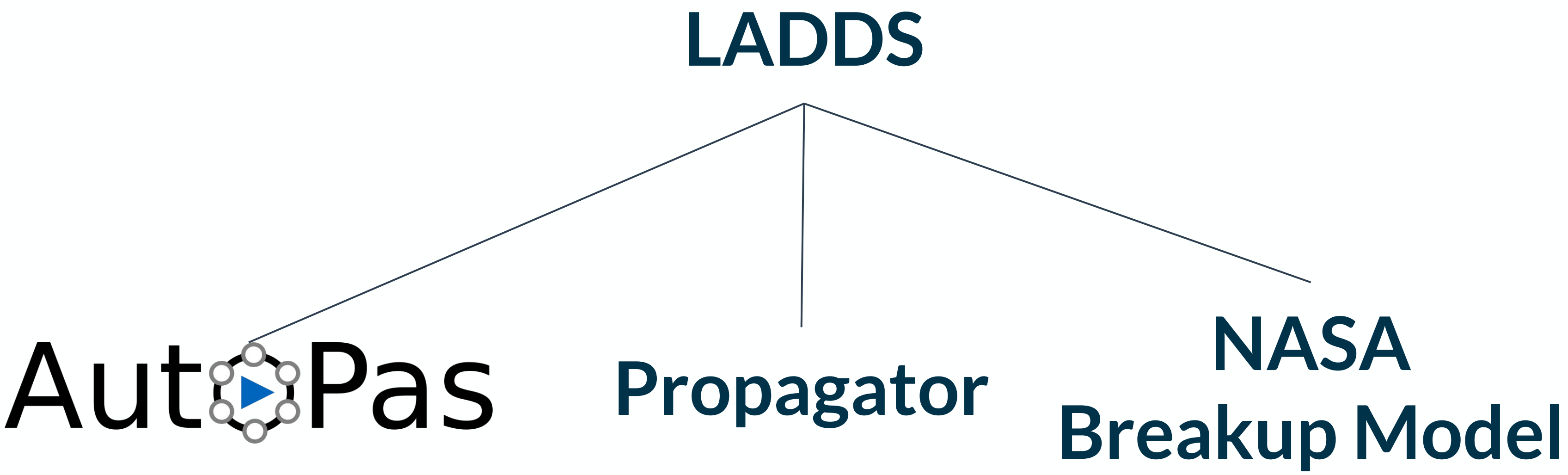
[1] Advanced Concepts Team, TEC-SF, European Space Agency

[2] Chair for Scientific Computing in Computer Science, Technical University Munich

Ariadna ID 21-5105

Contract Number: 4000134654/21/NL/AS

Produced Software Modules



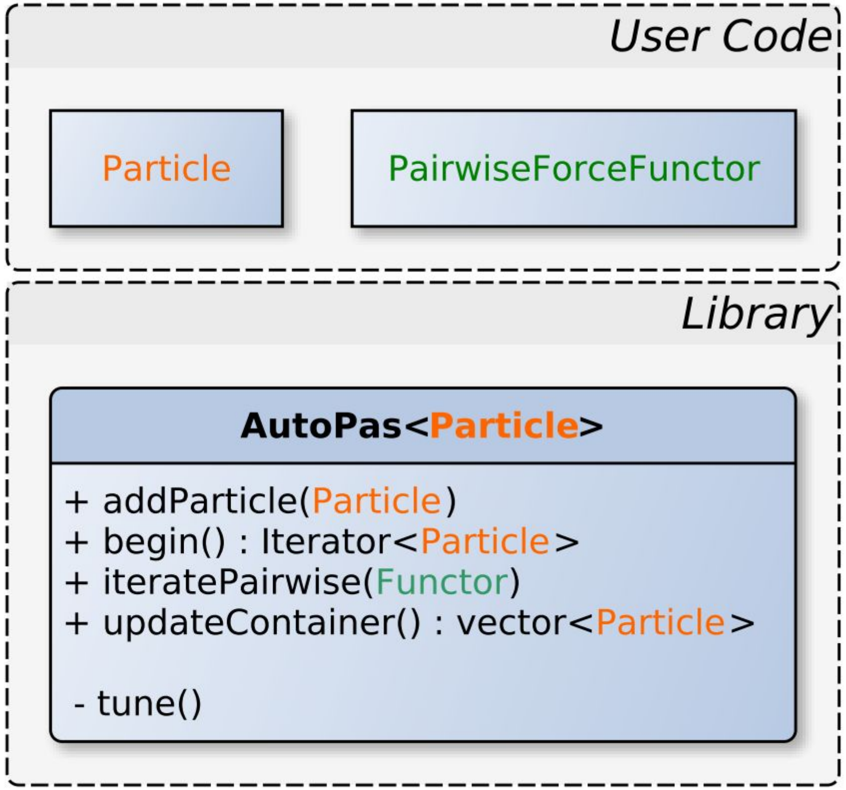
- Space debris simulator based on particle simulation
- Capable of massive parallelism
- Modular design with easily exchangeable components
- Use case: Testbed for different breakup models, propagators, initial conditions…

AutoPas

- Developed by TUM - SCCS
- Node-Level C++17 library
- Black-box particle container
- Facade-like software pattern
- User defines:
  - Properties of particles
  - Force for pairwise interaction
- AutoPas provides :
  - Containers, Traversals, Data Layouts, …
  - Dynamic Tuning at run-time

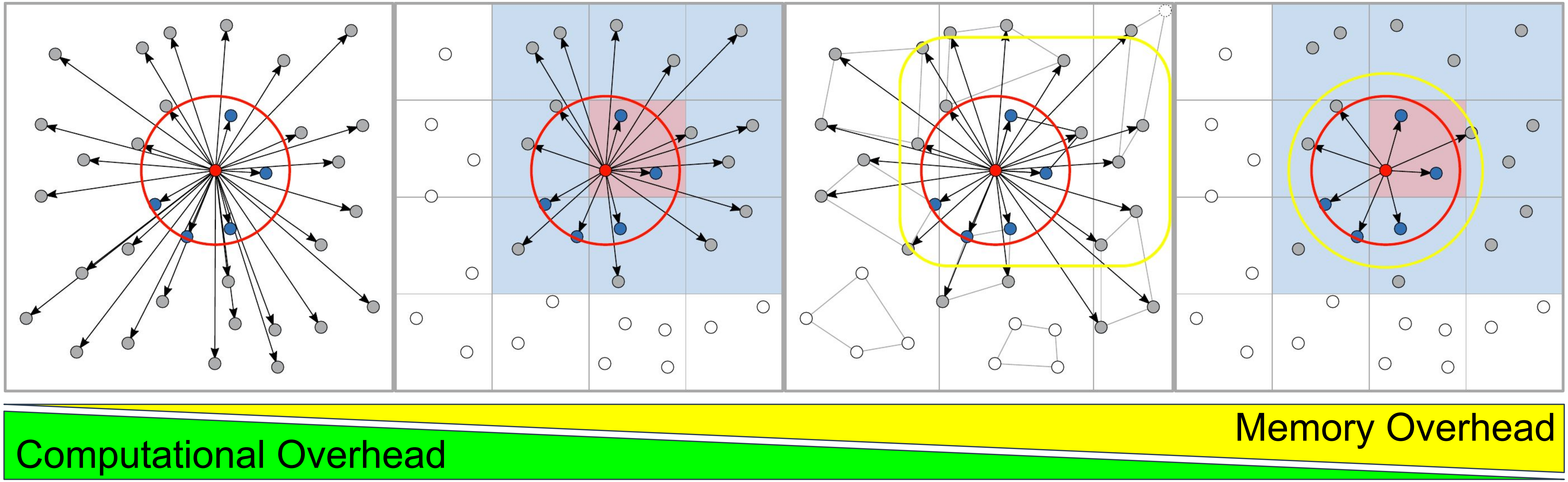
⇒ General base for N-Body simulations

Open Source: https://autopas.github.io

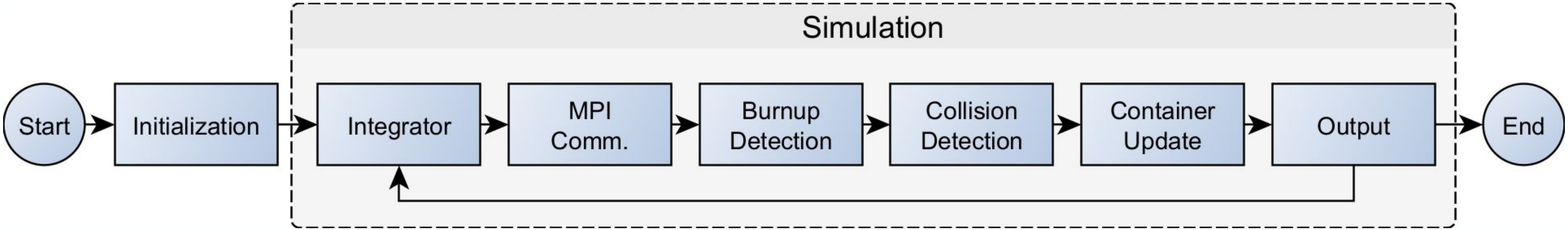


User Perspective

AutoPas Interaction Algorithms

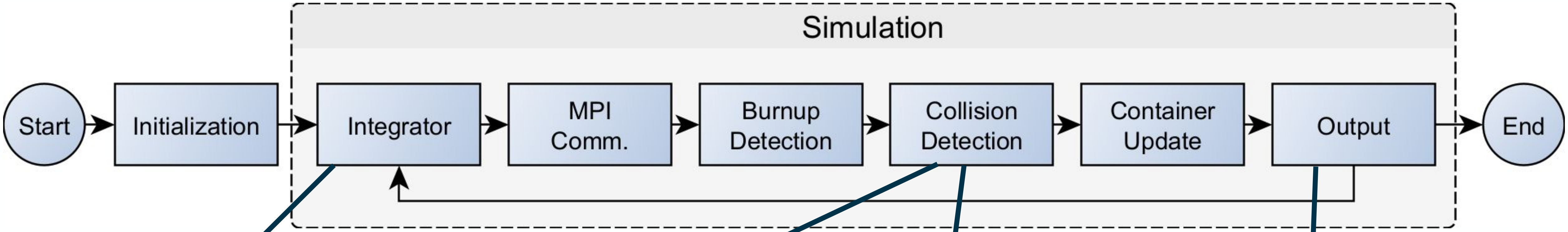


LArge-scale Deterministic Debris Simulation - 0.1.1



LADDS Simulation Flow

LArge-scale Deterministic Debris Simulation - 0.1.1



Implemented:
Symplectic 4th-order Yoshida

Prospective:
Higher-order symplectic, heyoka

Implemented:
Deterministic autopas detection

Prospective: Statistical Cubes Method

Implemented: NASA Breakup Model

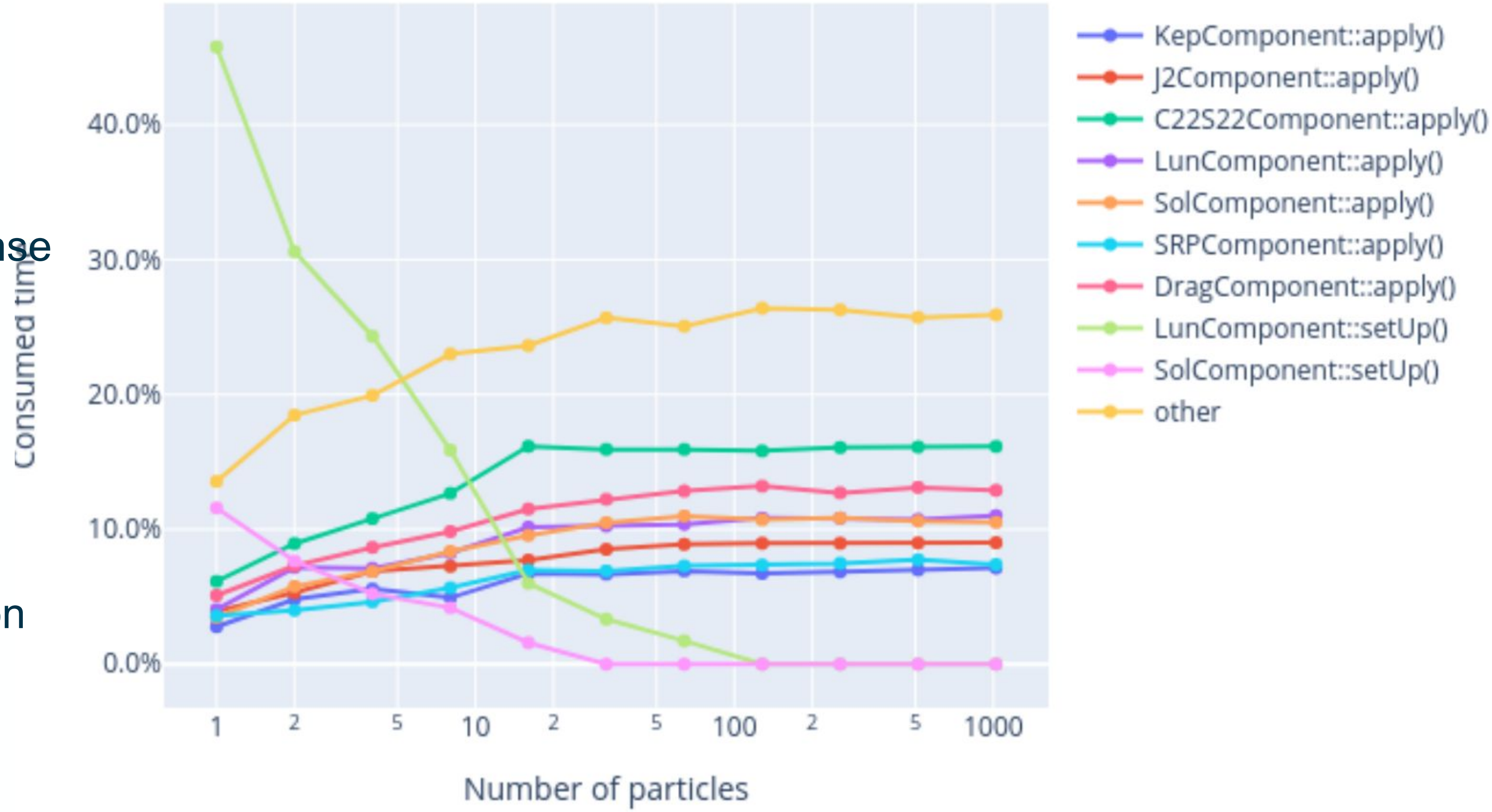
Prospective:
Higher-fidelity models

Implemented: HDF5, VTK, Checkpointing, Performance information

Prospective: Real-time visualization

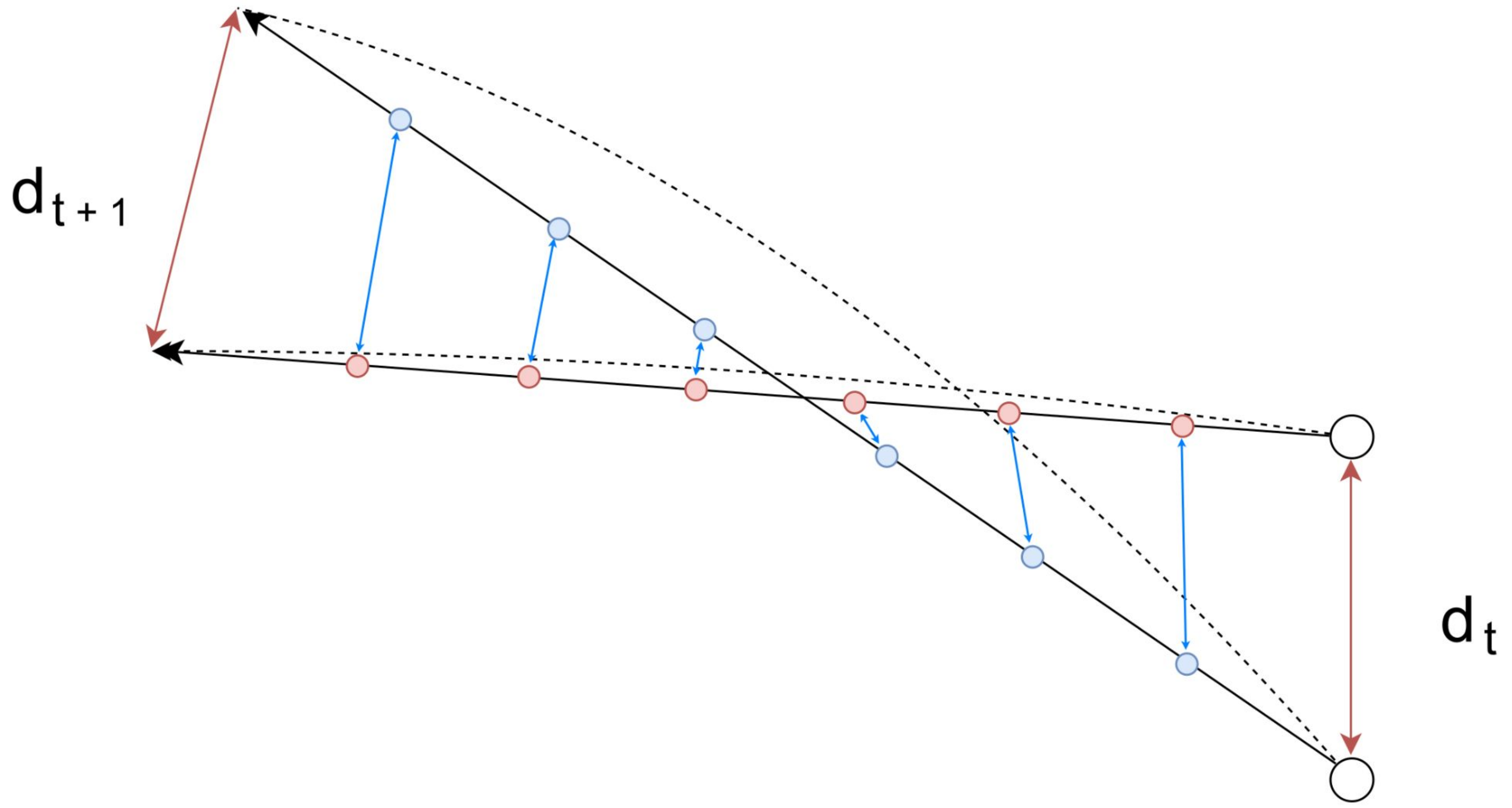
Yoshida Propagator

- Modern C++17
- Open-source under GPL-3.0 license
- Automated tests
- Validated against heyoka
- Extensive technical documentation



Bösing 2021

Conjunction Tracking



d_t - Distance at timestep t

Sub-timestep Interpolation

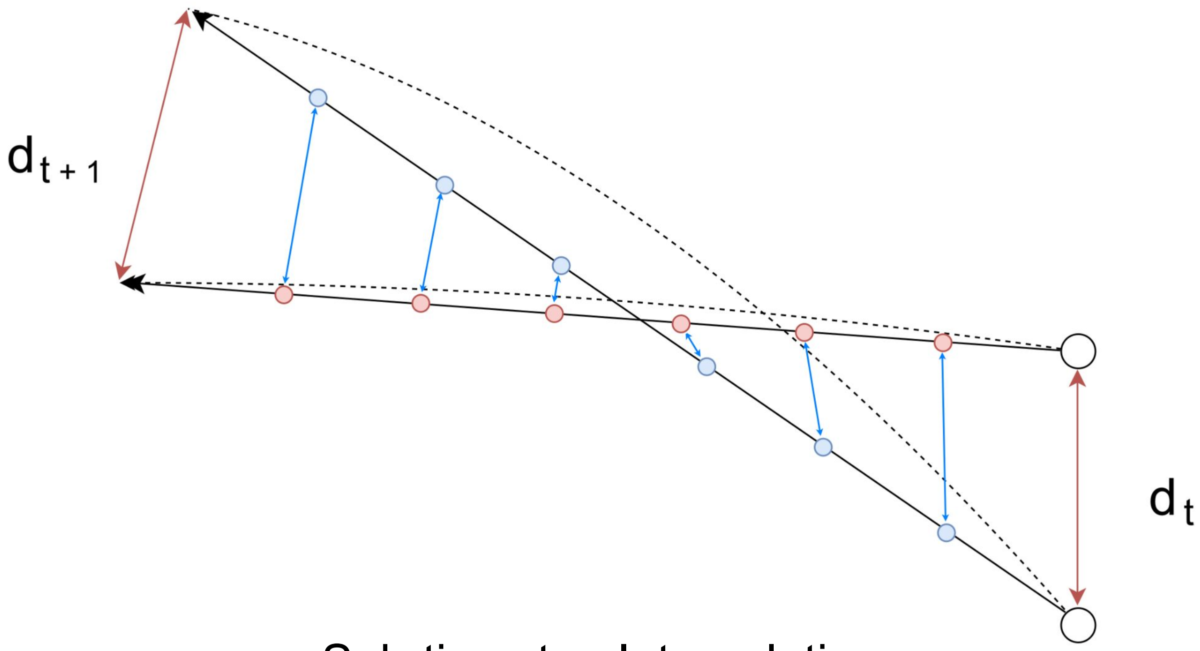
Conjunction Threshold κ

Collision

$$d \leqslant (r_1 + r_2)$$

Conjunction

$$d \leqslant \kappa(r_1 + r_2) \text{ with } 0 < \kappa \leqslant 10$$



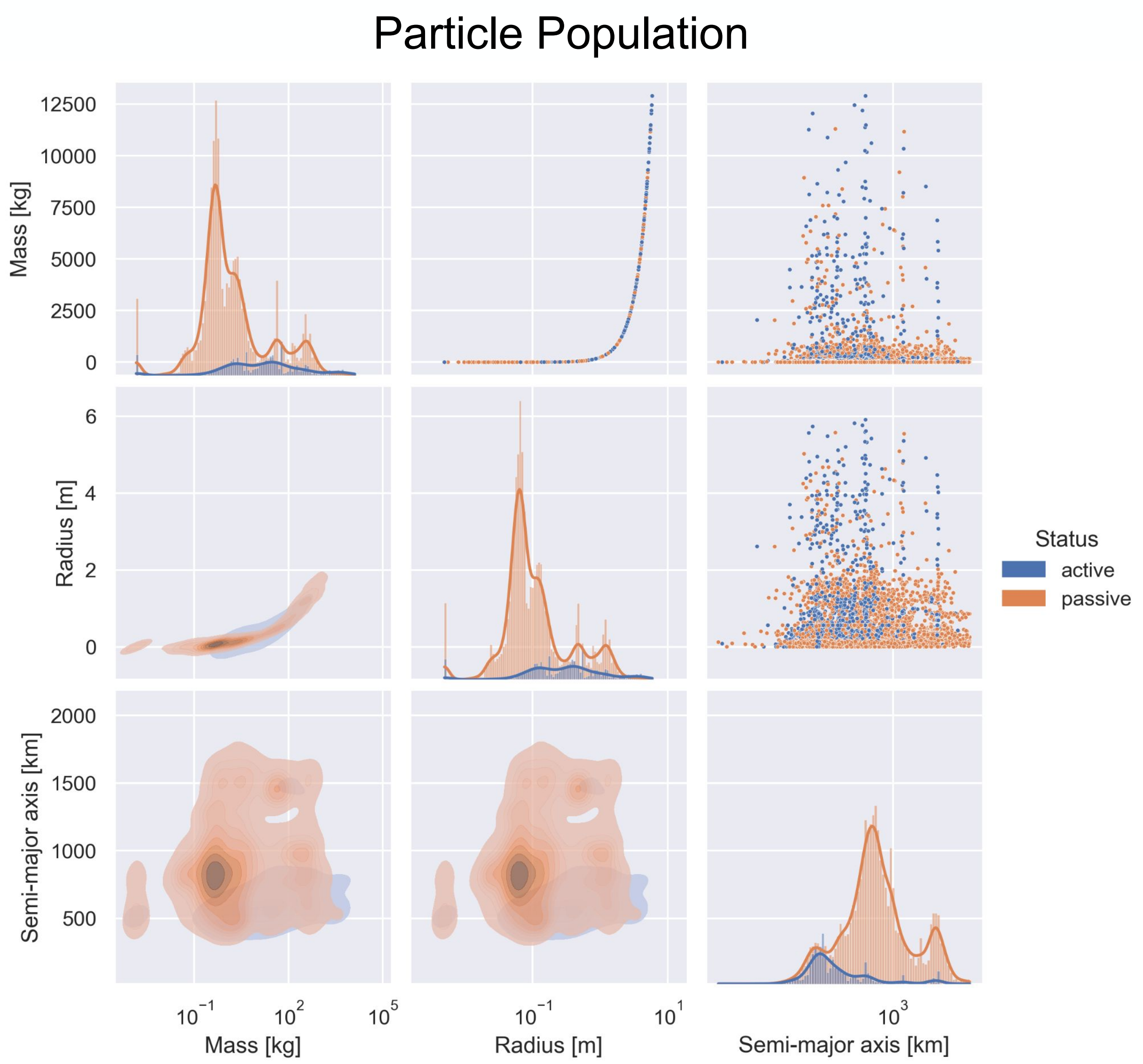
Sub-timestep Interpolation

d_t - Distance at timestep t

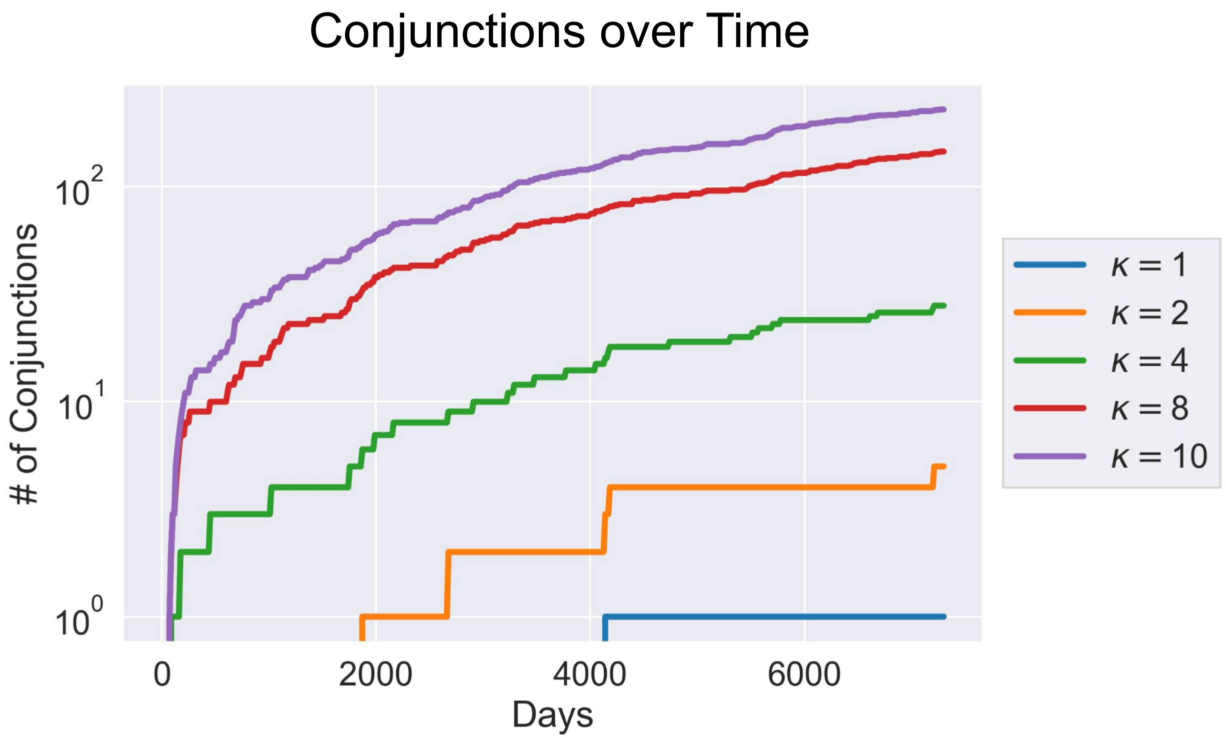
r - Particle radius

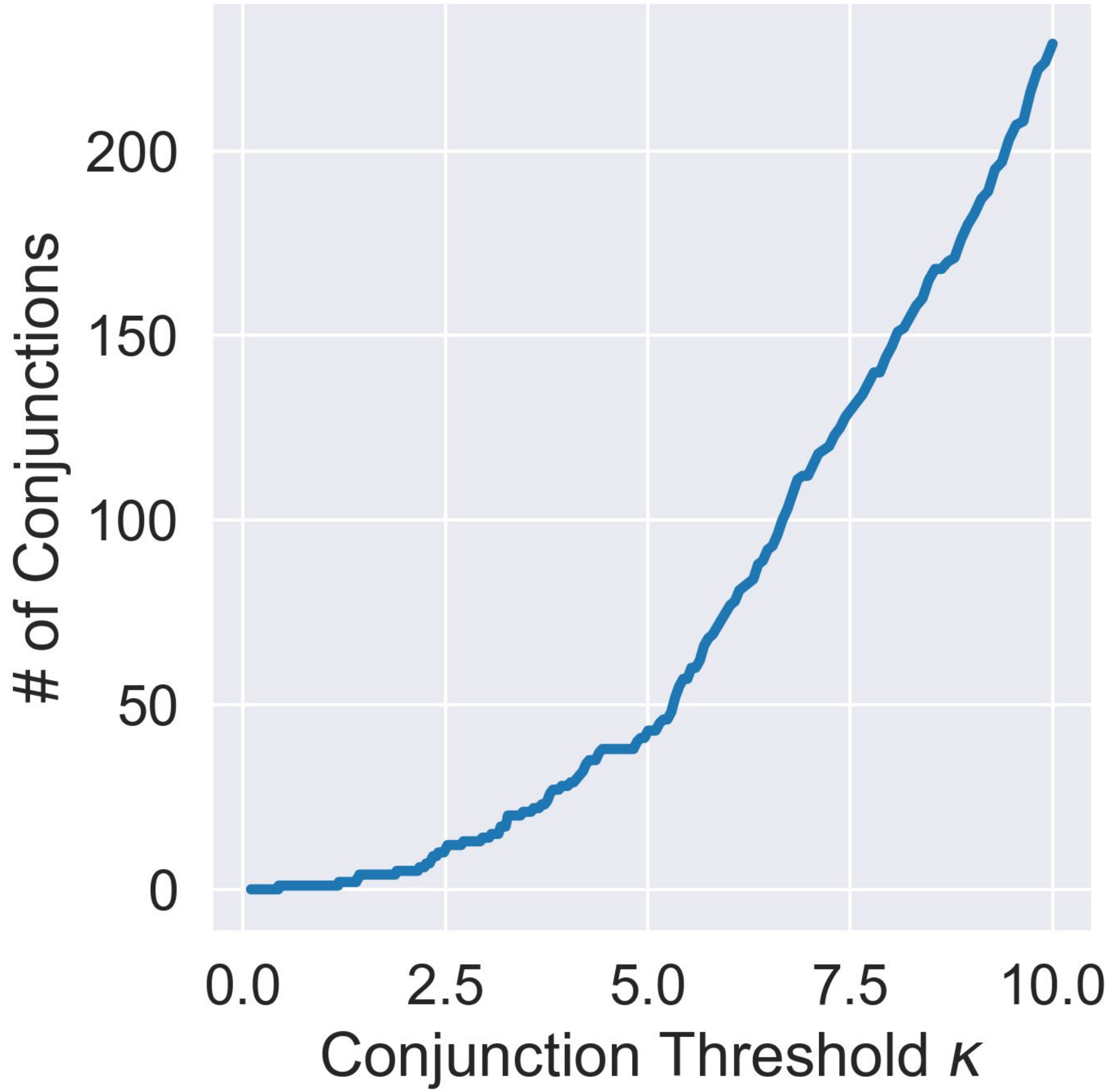
Simulation Setup - The ≥ 10cm case

- Data from *CelesTrak* and *space-track.org*
- 16 024 objects between 175 and 2000 km
- Differentiate between 1996 active and 14 028 passive objects
- Results are on a single-node / 28 threads at LRZ
- 20 years simulation time, timesteps of 10 seconds executed over 9 days (218h)
- Symplectic integrator accounting for: Keplerian motion, spherical harmonics, solar radiation pressure, drag



Conjunctions - Few Collisions, More Conjunctions



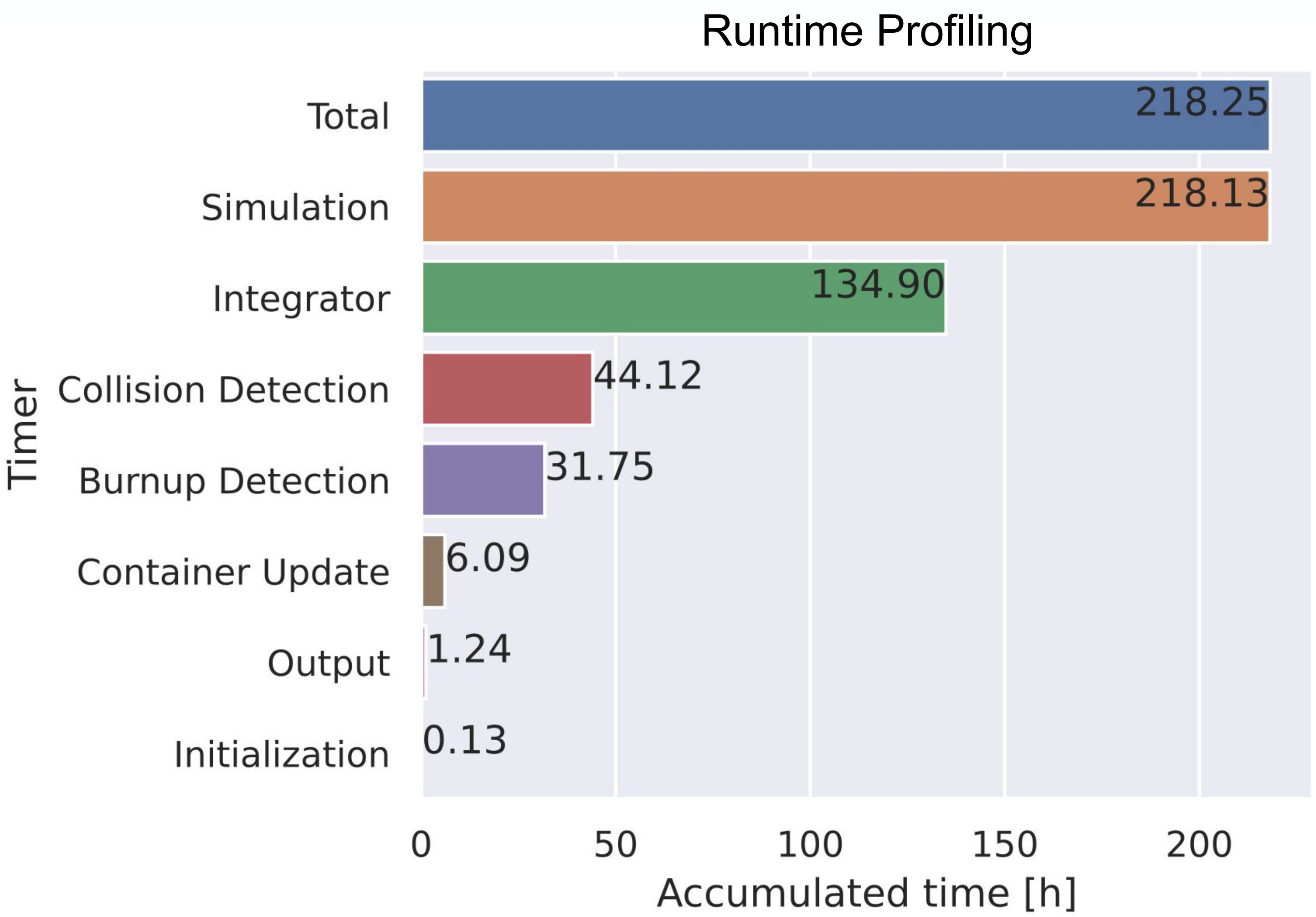


Single Node Performance

- ~9 days runtime on relatively slow CPUs
- AutoPas enables efficient conjunction tracking
- Integrator / Propagation are bottleneck (but embarrassingly parallel)

↓

- Enormous potential for scaling up with MPI
- Burnups have room for optimization (fixed by now)

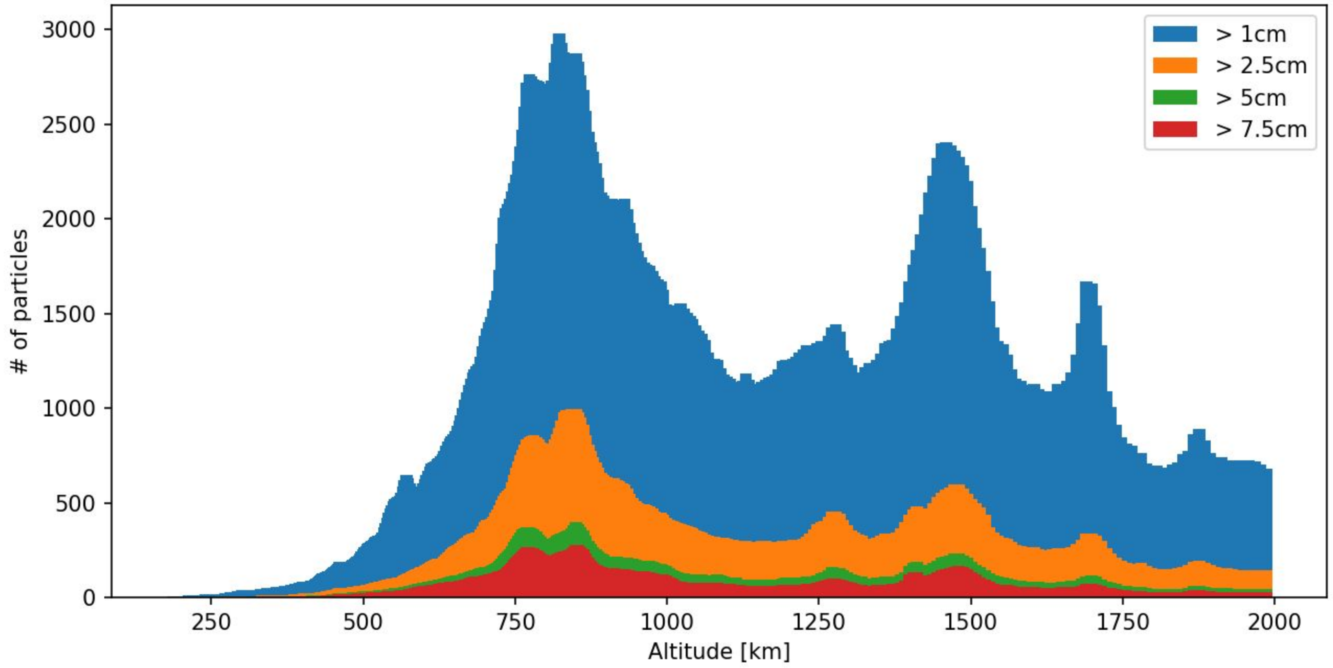


Simulation Setup - The 1 to 10 cm population

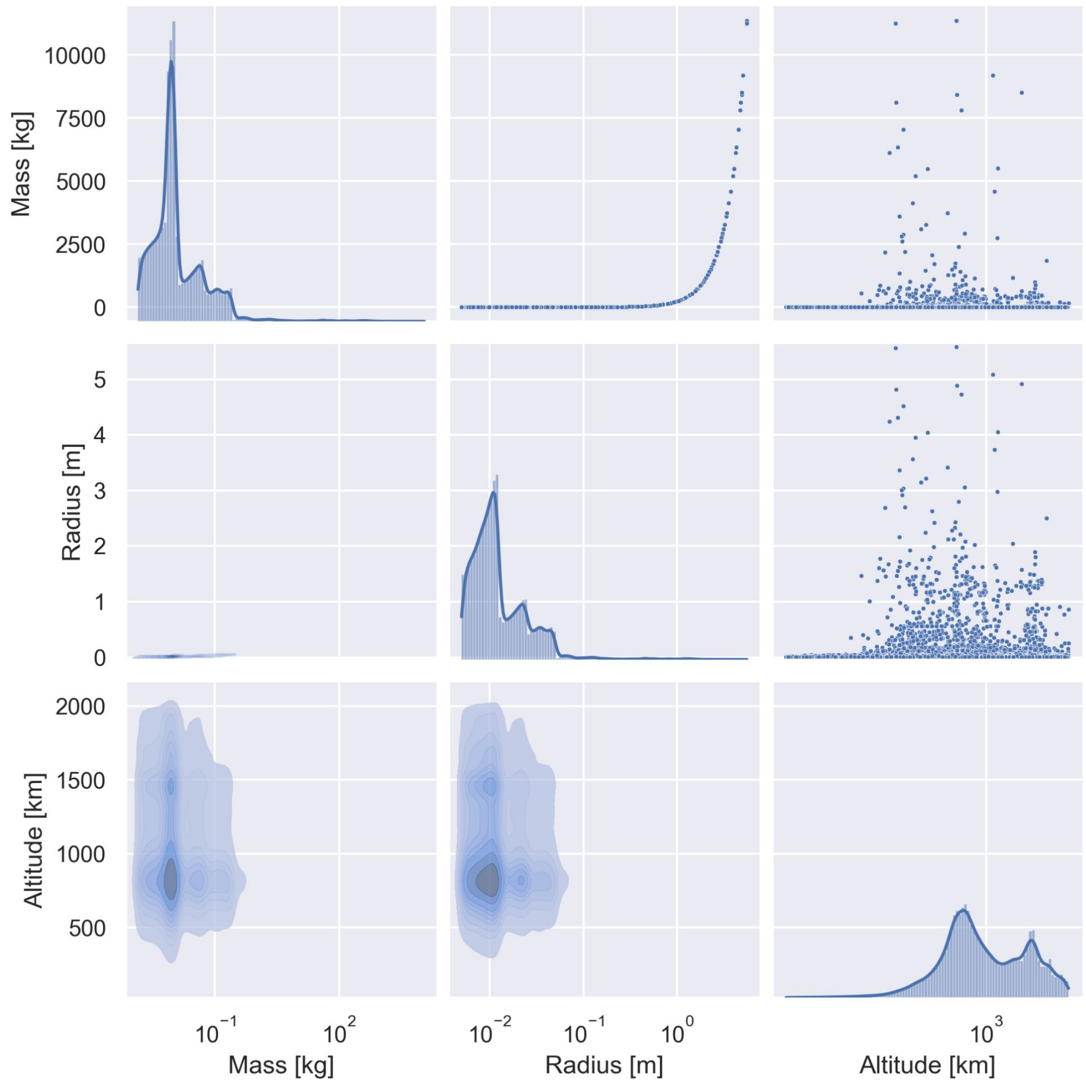
- Based on MASTER-8, four size bins, 1000 altitude

- 16 024 large (>10cm)
  + 598 491 small passive objects

- Inclination & eccentricity as base population

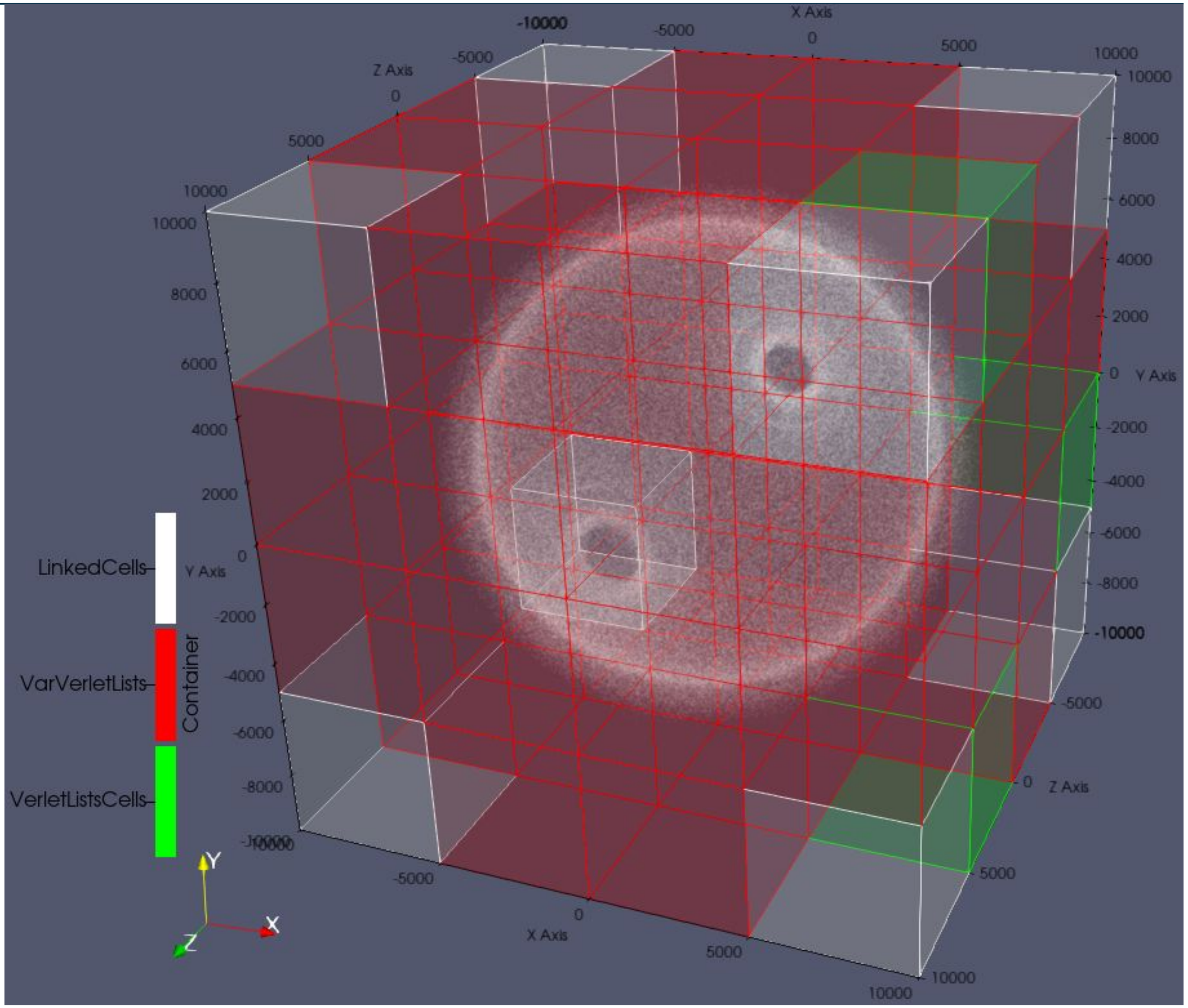


Both Particle Populations



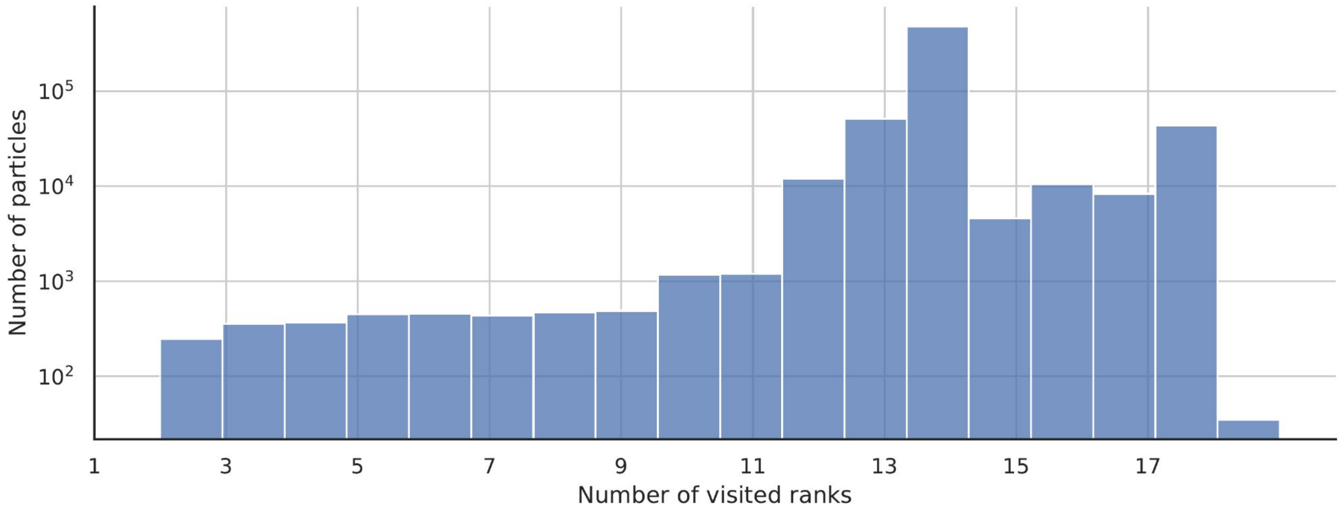
Large Scale 1 - Naive Grid

- Classic Molecular Dynamics approach: divide domain into a regular grid
- Easy and quick to implement
- One AutoPas instance per subdomain
- Optimized algorithm for every subdomain
- Fitting cuboids to a sphere is not optimal…



Naive Grid - Suffers from Communication

- Particles traverse many ranks
- Induces lots of communication
- Need for a decomposition that follows trajectories

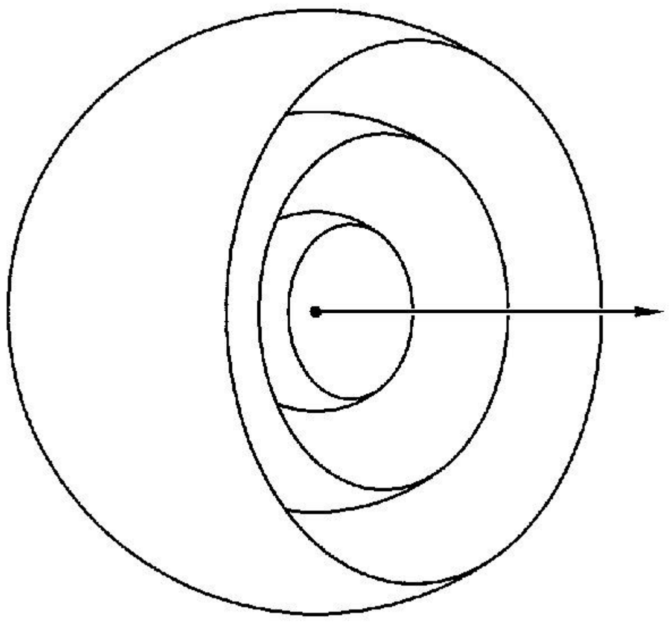


Number of visited ranks per particle over 800 time steps

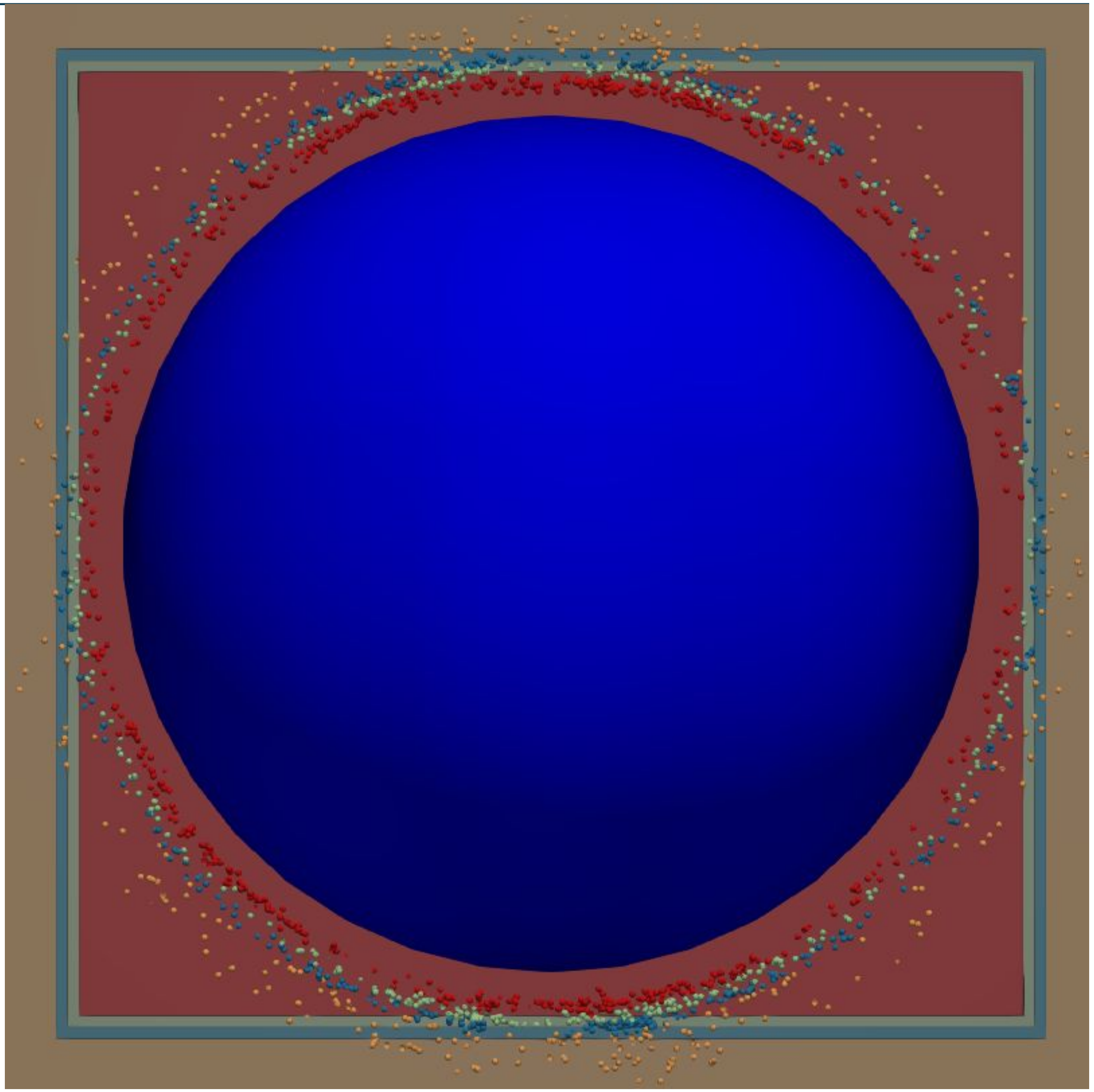
Large Scale 2 - Altitude based

- Problem: AutoPas supports only cuboid domains

- Solution: Nested cubes for altitude ranges



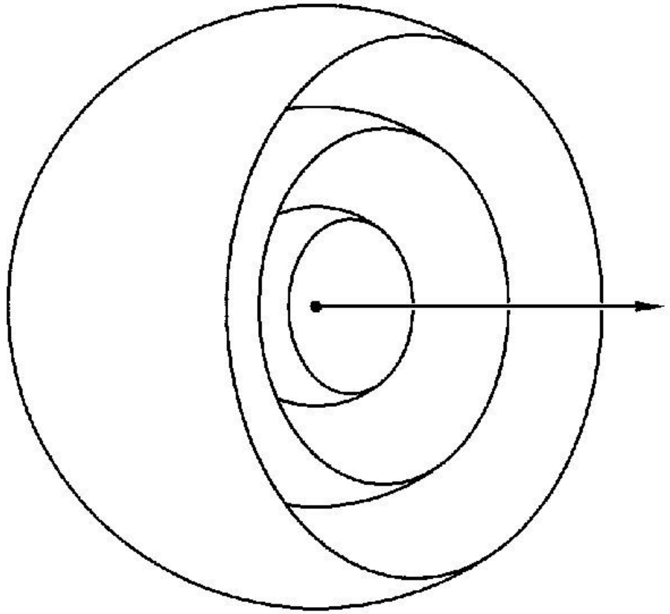
Littlejohn & Rensch 1997



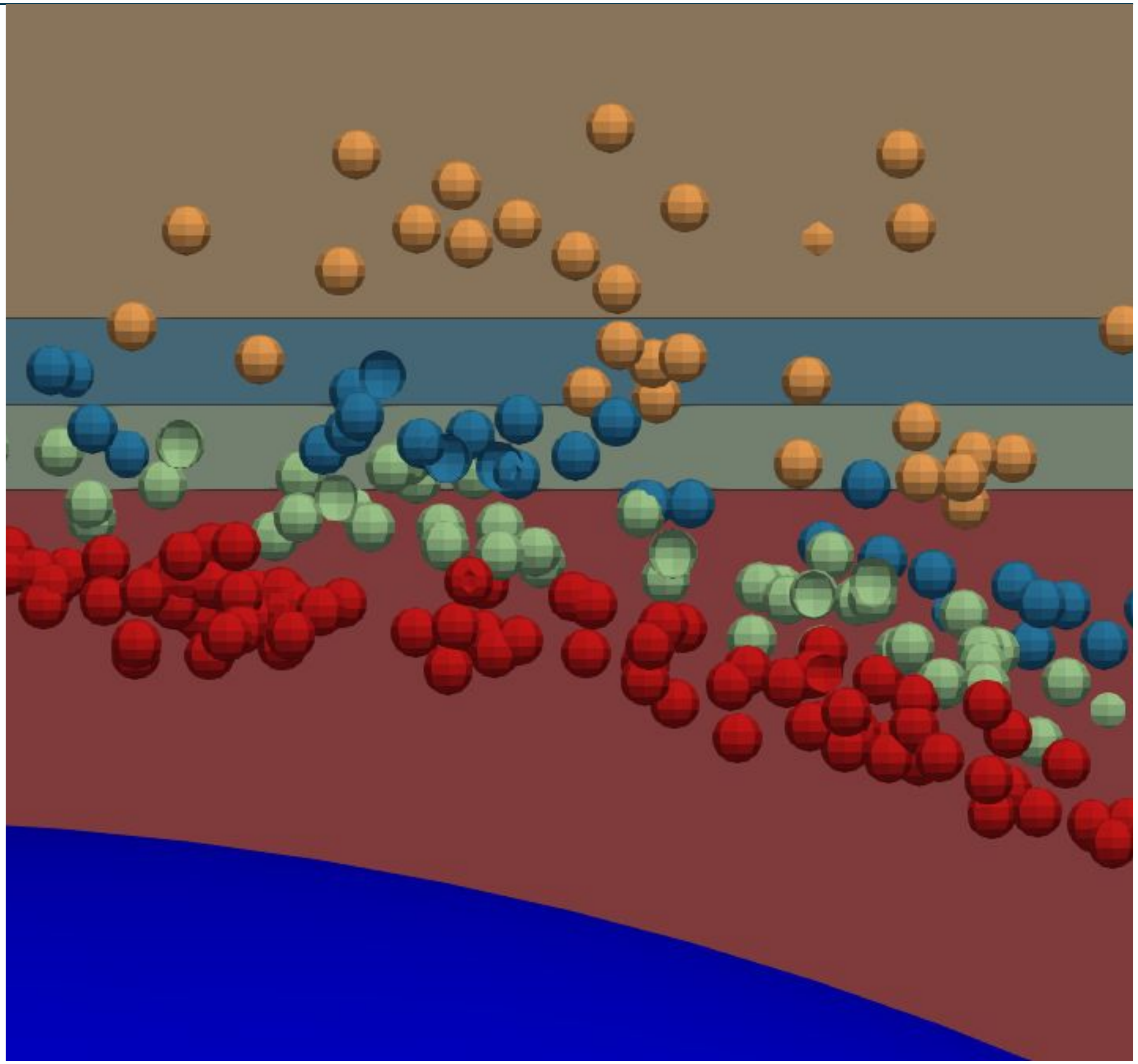
Large Scale 2 - Altitude based

- Problem: AutoPas supports only cuboid domains

- Solution: Nested cubes for altitude ranges

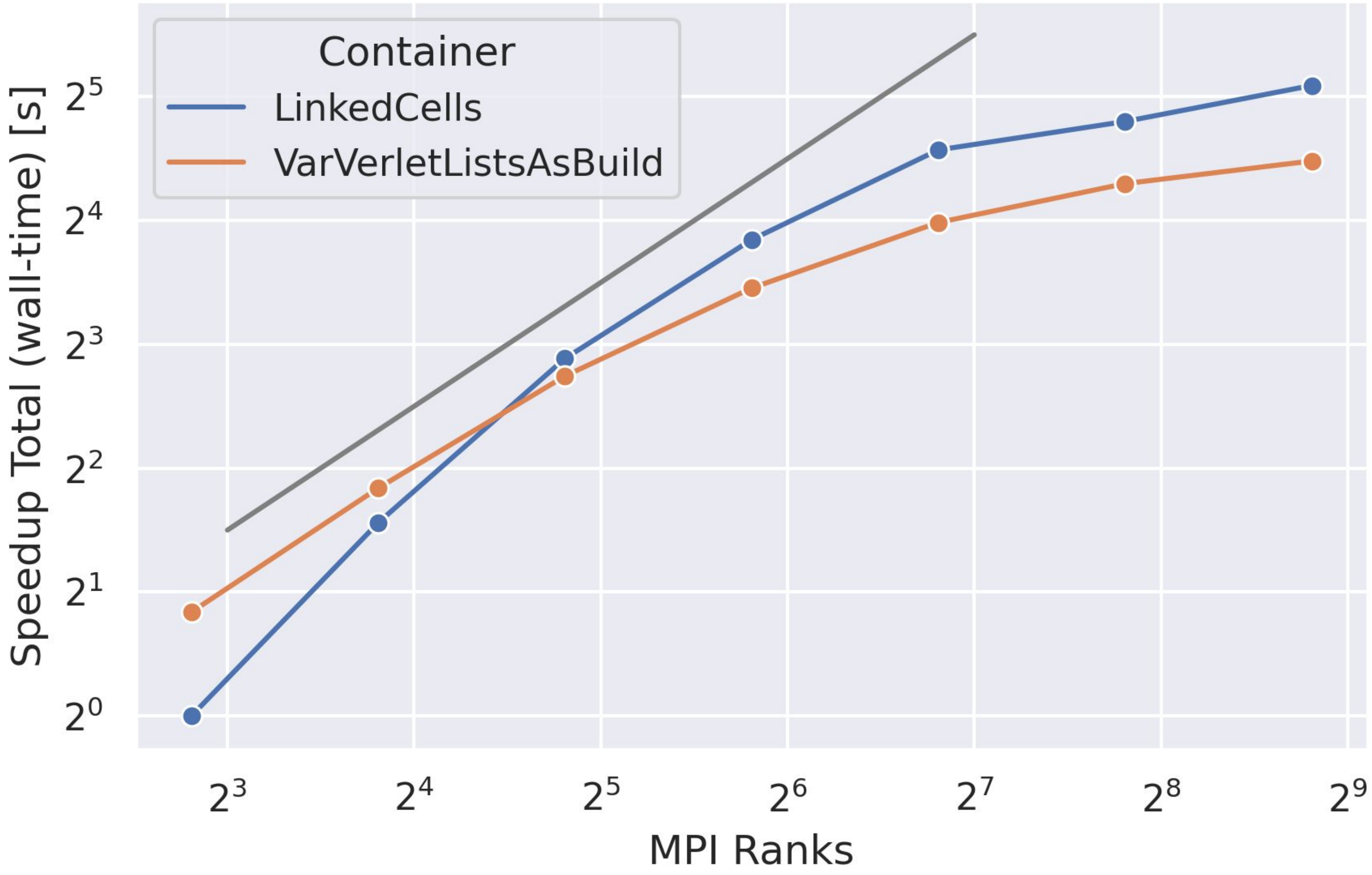


Littlejohn & Rensch 1997



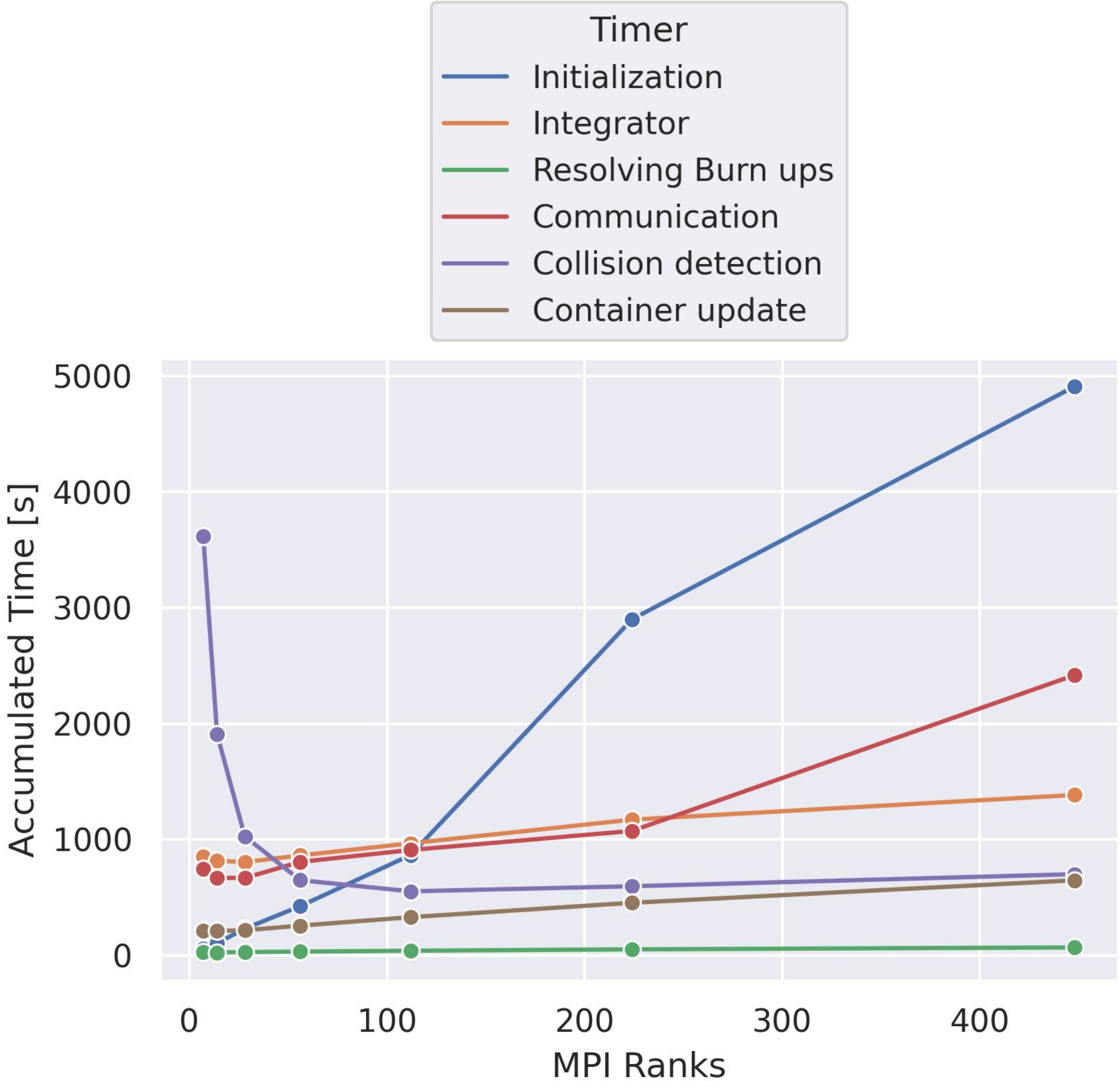
Strong Scaling - The full picture

- 614 515 Particles
- 7 - 448 MPI Ranks
- For large runs Linked Cells beat Verlet Lists



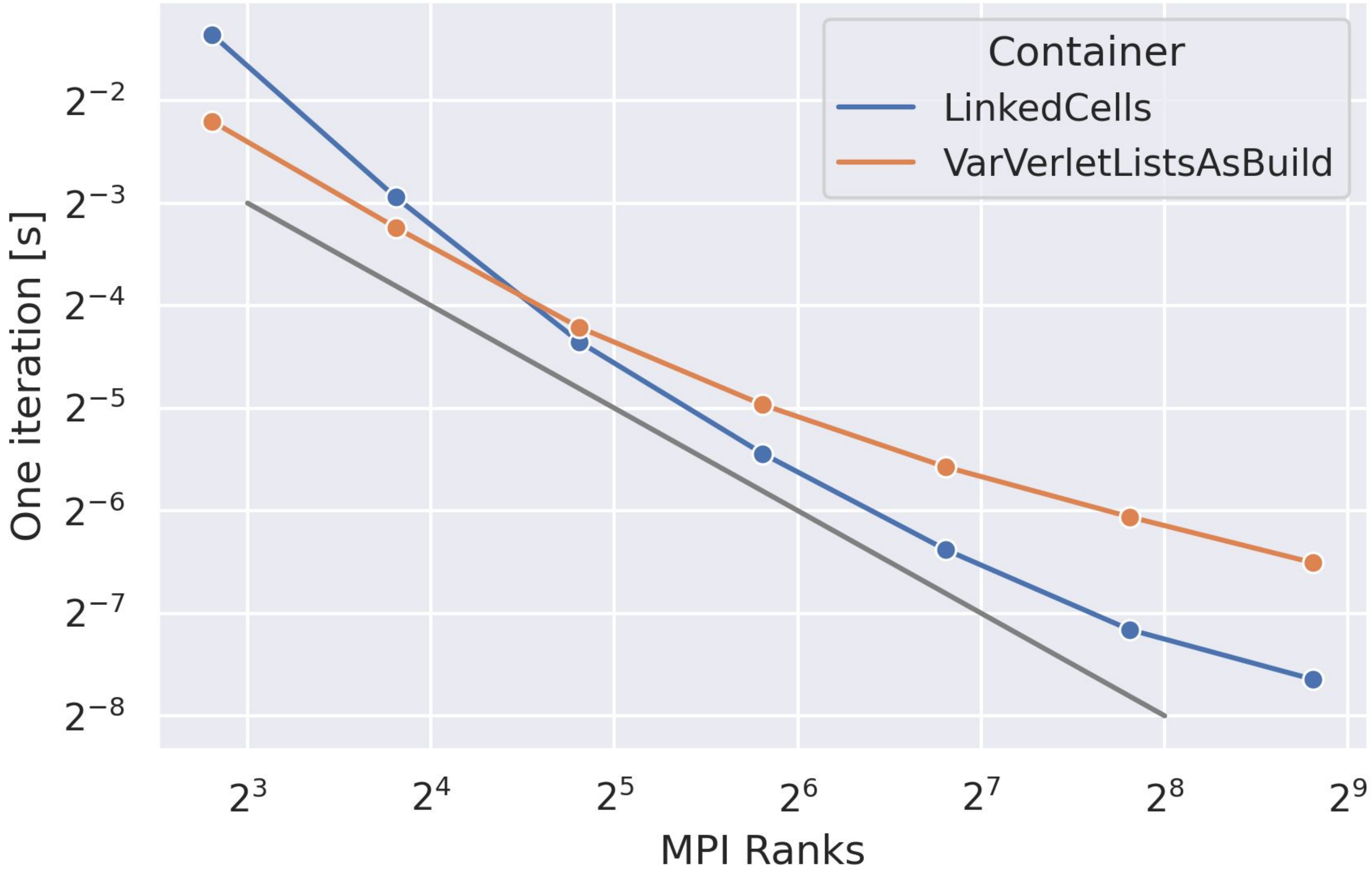
Where are potential Bottlenecks?

- Initialization grows fastest
- Communication only starts to rise for a high number of ranks
- Collision detection becomes cheaper since fewer neighbors are considered



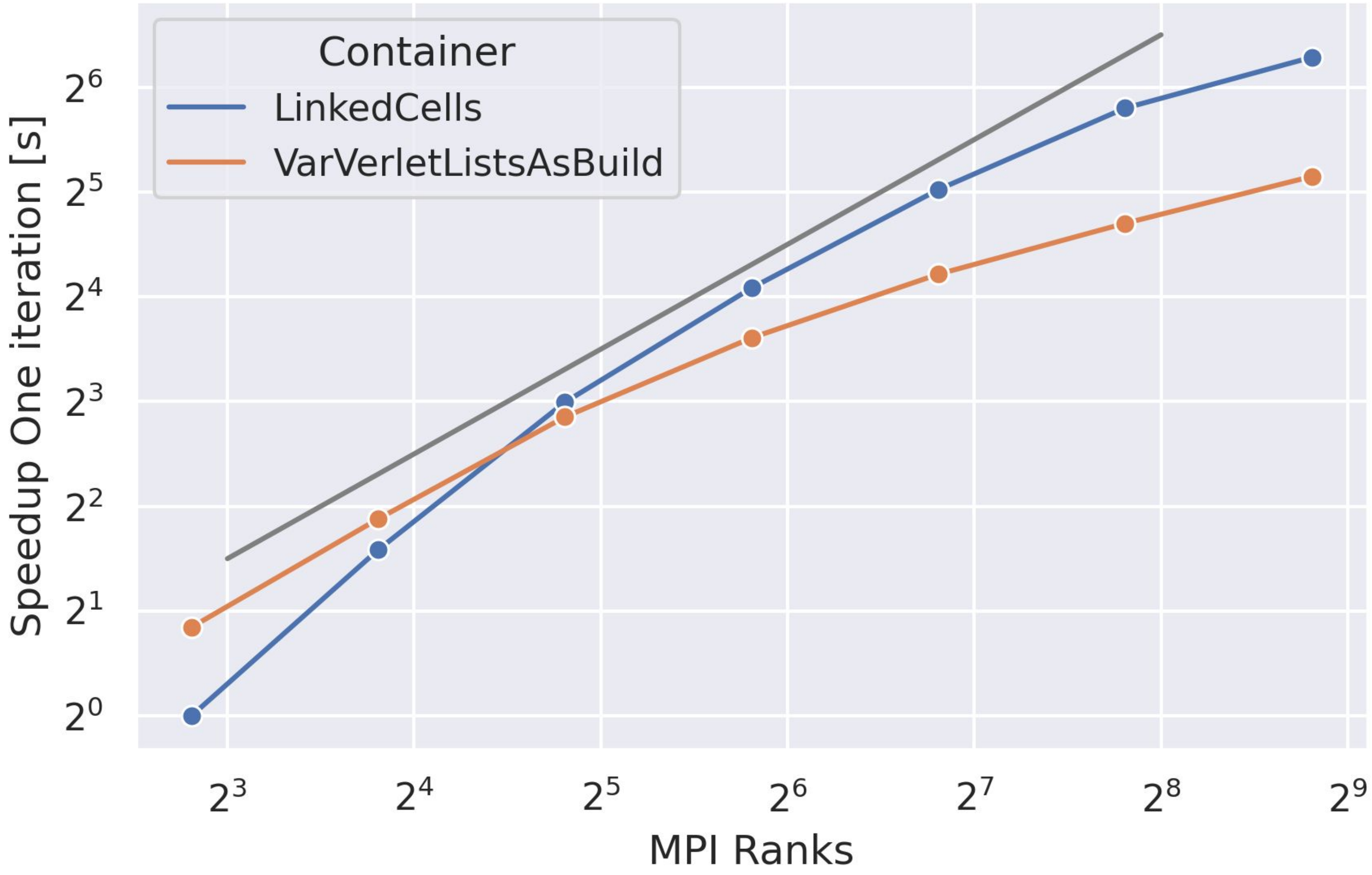
Strong Scaling - Iteration time

- Short benchmark with 1000 iterations
- Significantly better scaling during production iterations

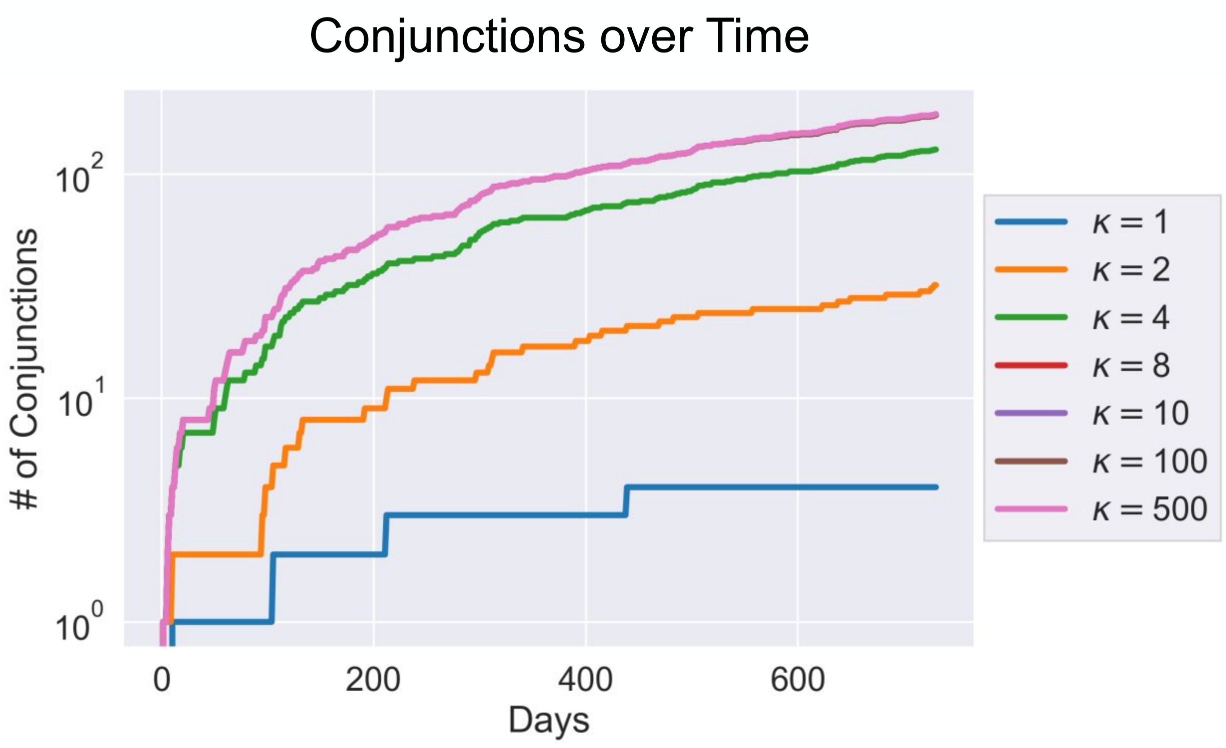


Strong Scaling - Scalability

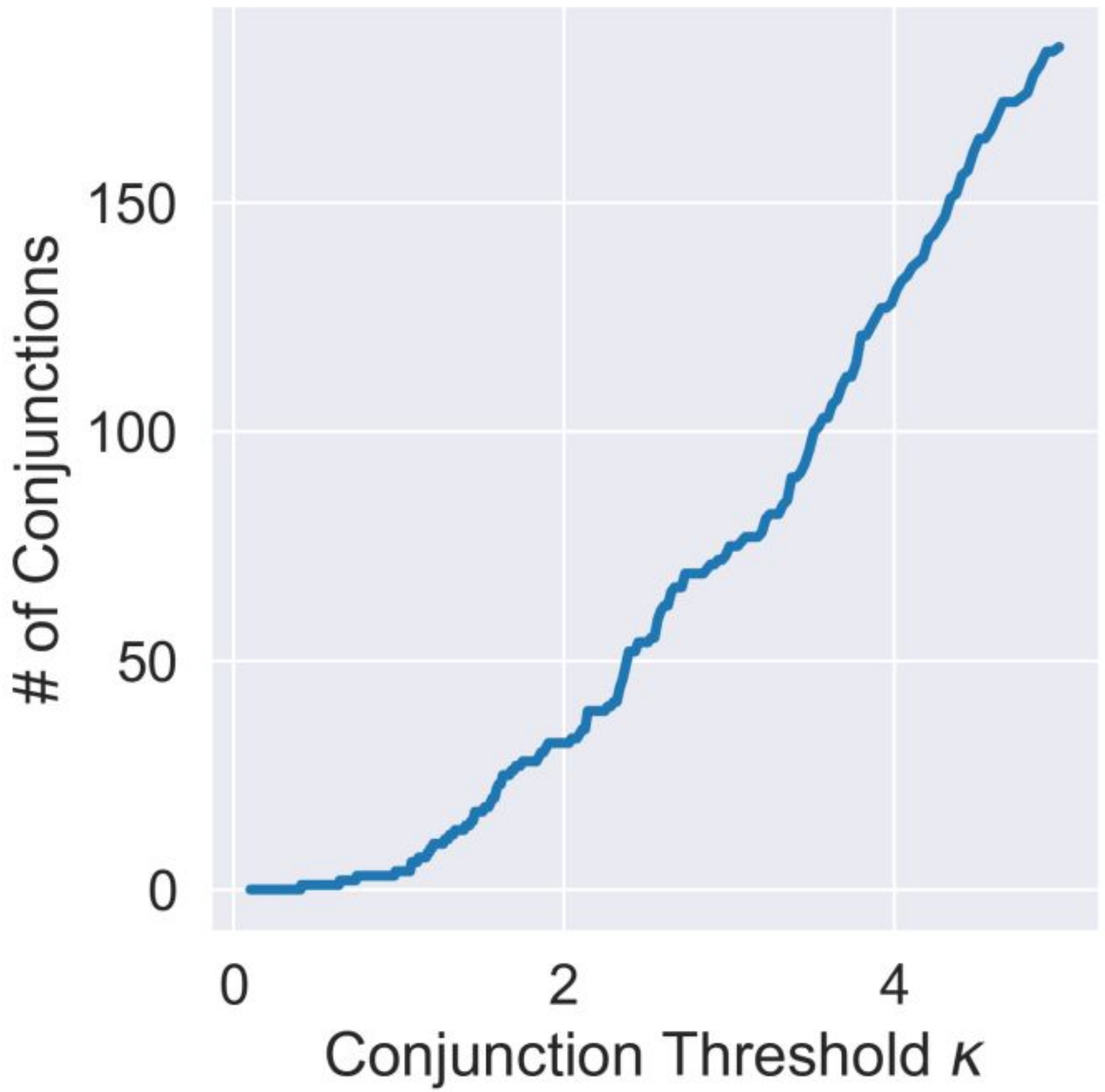
- Very good scaling at least until 400 ranks
- At 448 ranks only 1371 particles per rank
- Slightly superscalar scaling in the beginning



Conjunctions - 600k Population

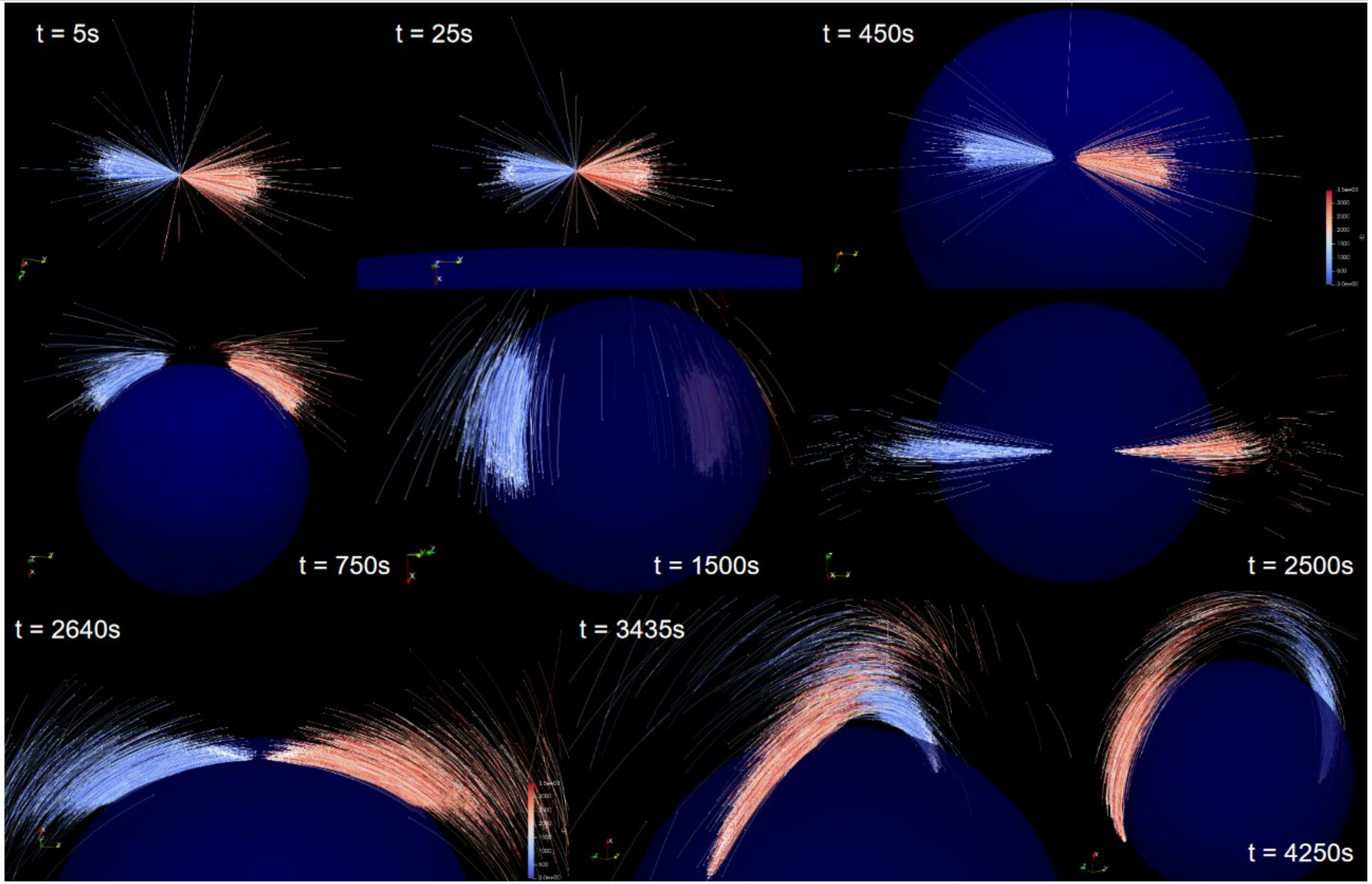


- Two years (6.3 Mio time steps)
- Simulation time 8.48 hours
- 448 MPI Ranks using 1792 Compute Cores



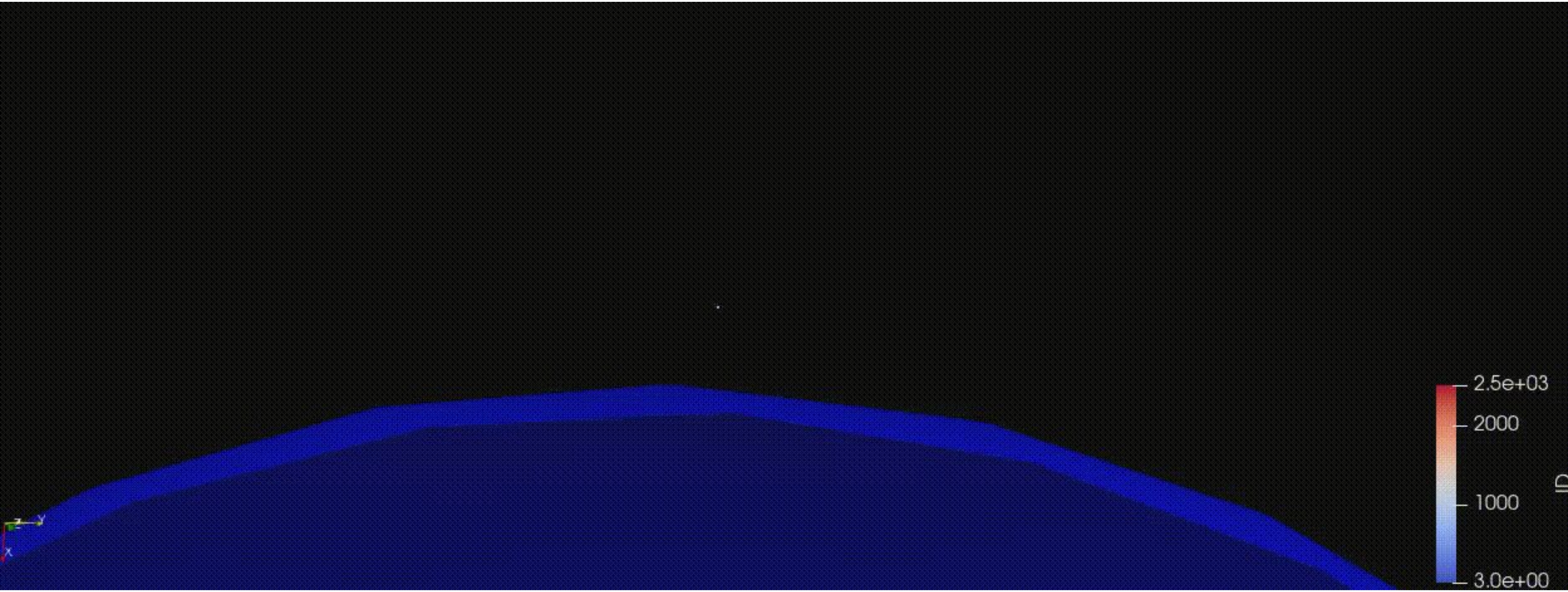
NASA Breakup Model

- Modern C++17
- Open-source GPL-3.0 license
- Validated against existing implementations
- Automated tests
- Extensive technical documentation



Exemplary Break-up Event

LADDS + NASA Breakup Model in Action



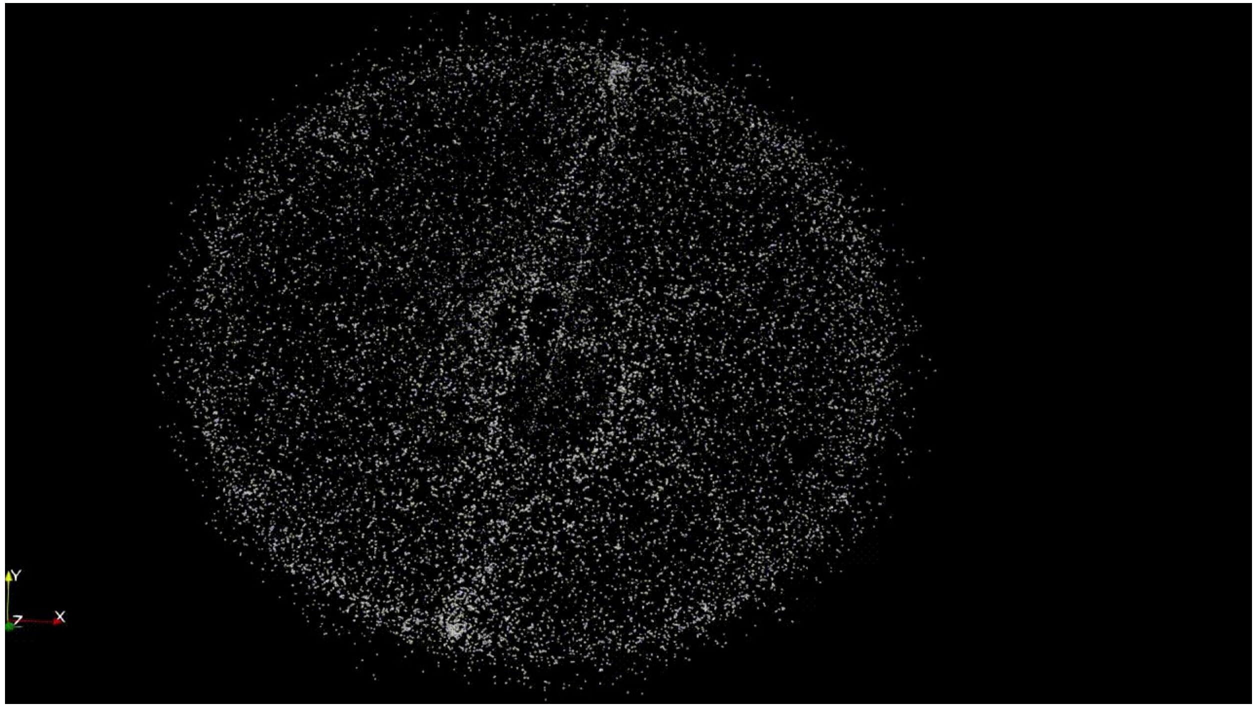
Conclusion & Outlook

- All study objectives fulfilled
  - Demonstrated **feasability** of simulating debris non-stochastically over long time ranges
  - Demonstrated **scalability** for **>600k** particles on up to 1792 cores

- **~50x real time**

- Optimizing MPI communication is key

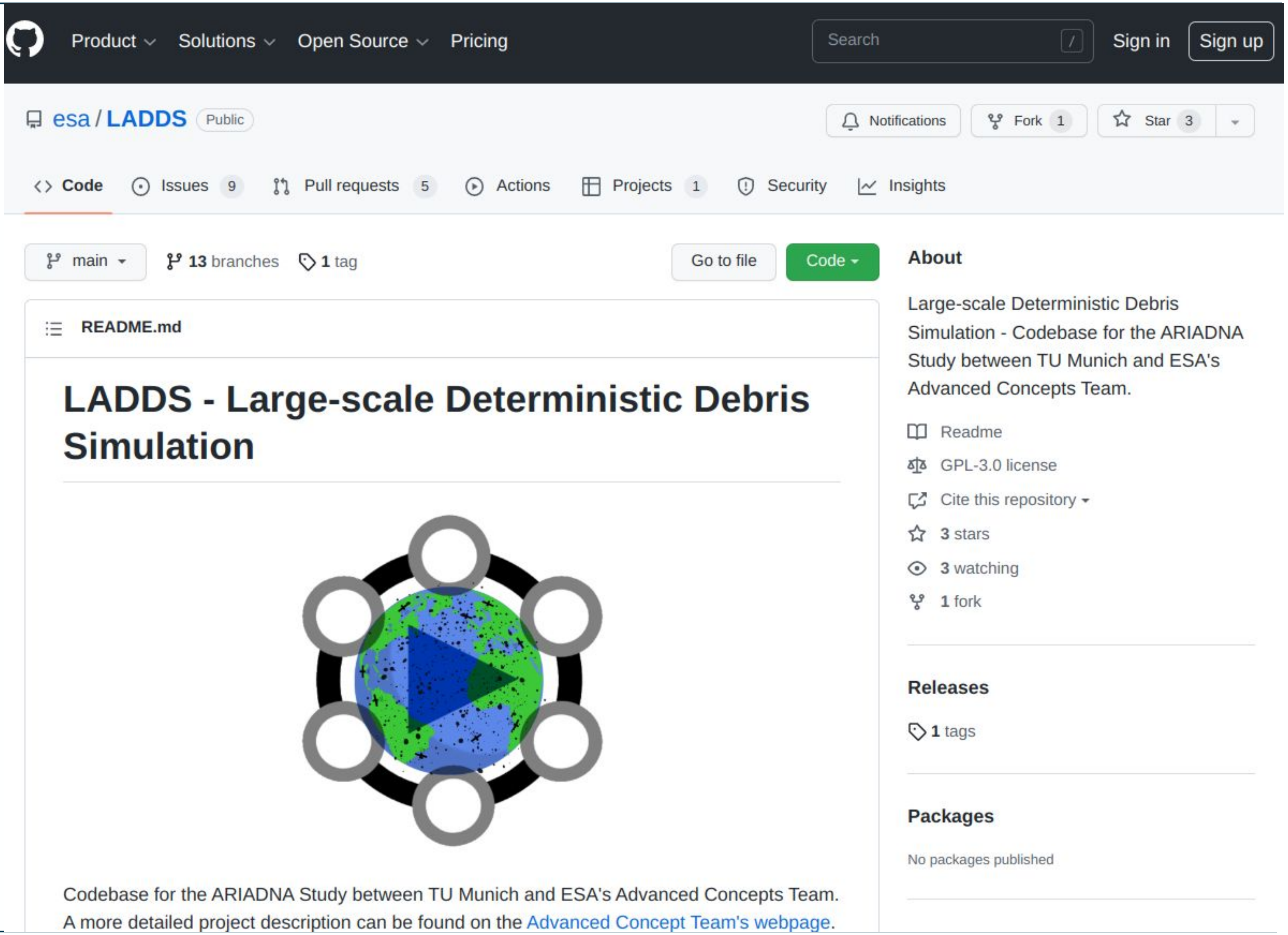
- Created **open source** simulation framework
  - LADDS
  - Yoshida Propagator
  - NASA Breakup Model



Visualization of LEO Objects

Try LADDS yourself

- Publicly available on GitHub
- Up to date documentation
- Automated Unit tests (CI)
- Automatic dependency handling
- Easy configuration via YAML



Thanks!

Code open-source available

https://github.com/esa/LADDS/

https://github.com/FG-TUM/OrbitPropagator/

https://github.com/esa/NASA-breakup-model-cpp

https://github.com/AutoPas

Questions?