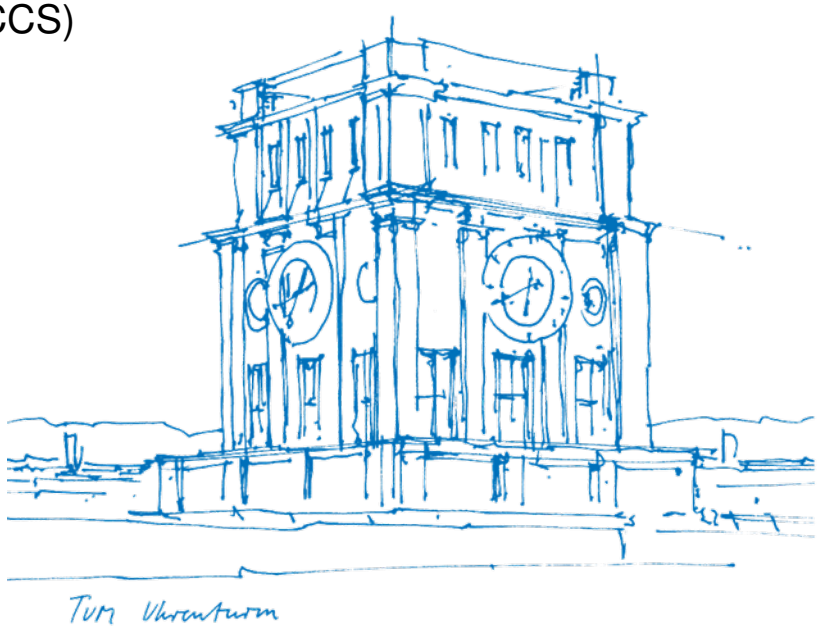
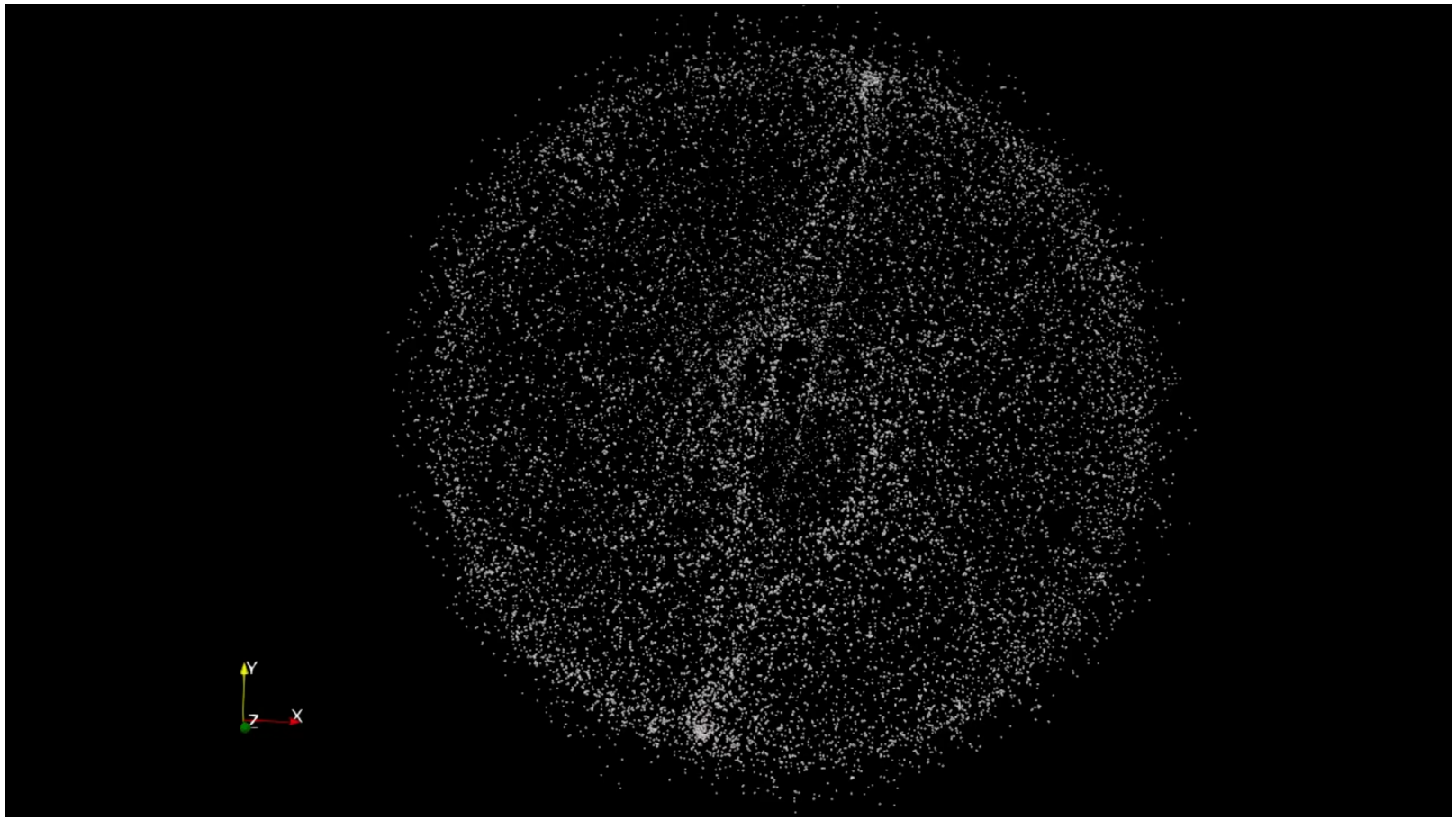
Pimp My Simulation Further:

Combining Strategies for Dynamic Algorithm Selection for Particle Simulations

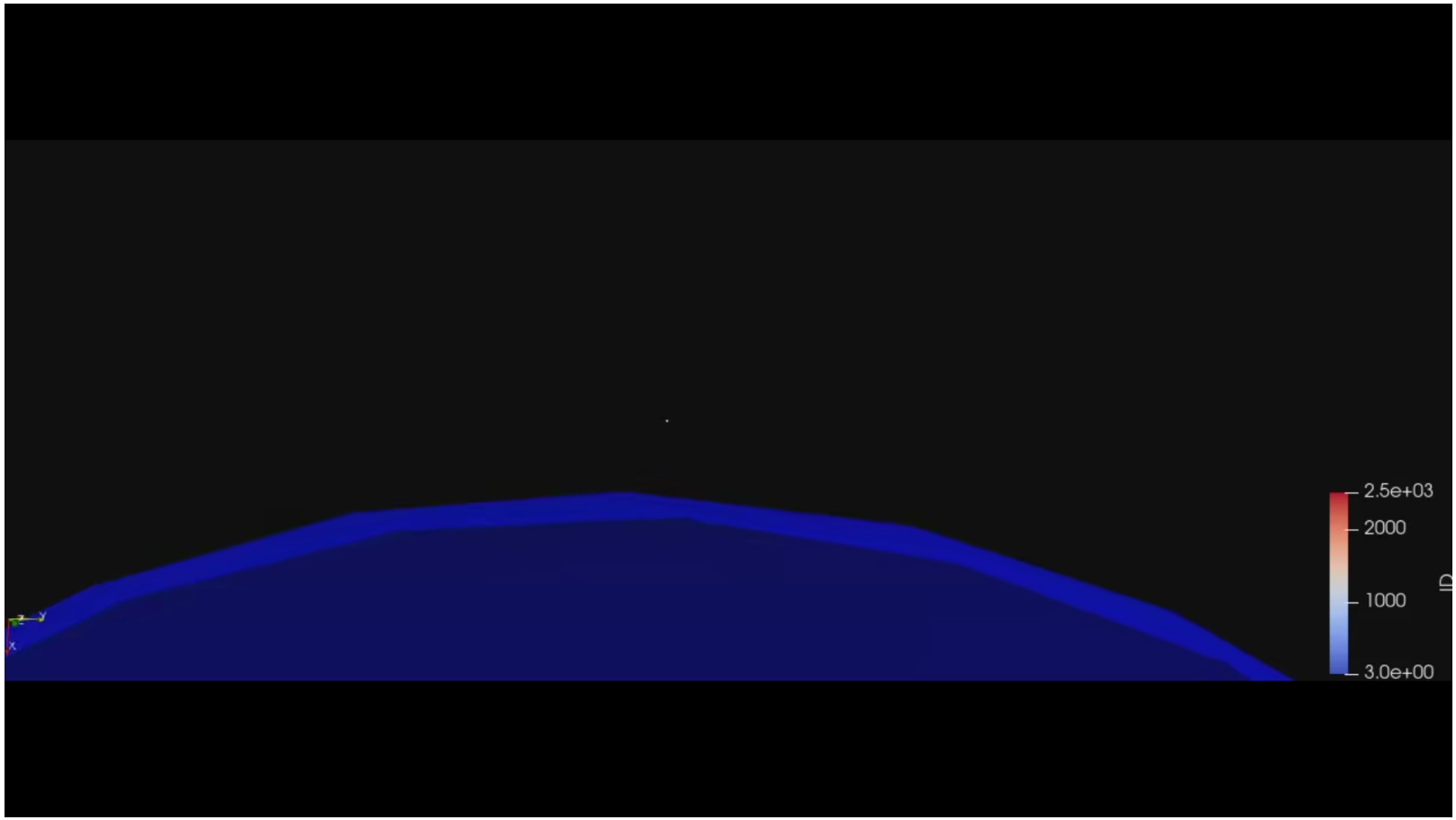
Fabio Gratl, Hans-Joachim Bungartz, Philipp Neumann
Technical University of Munich
TUM School of Computation, Information and Technology (CIT)
Chair of Scientific Computing in Computer Science (SCCS)
Leogang, 13.03.2024



Motivation: Space Debris

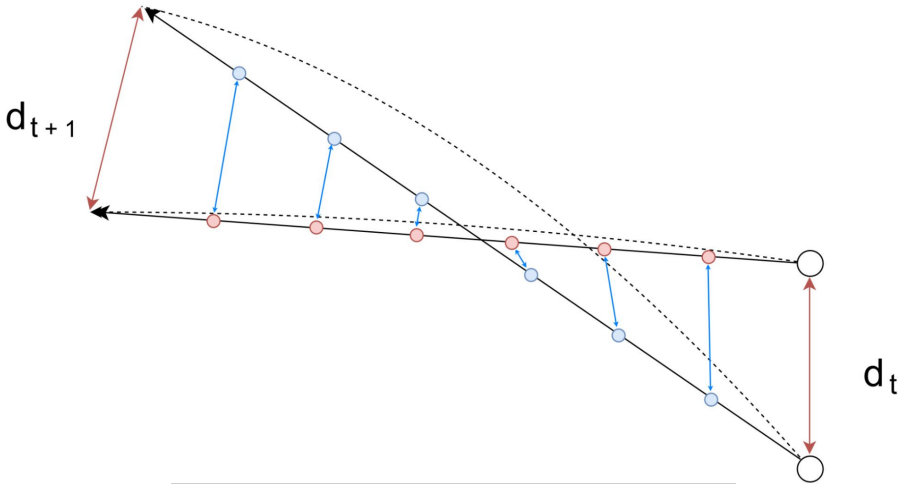


Motivation: Space Collisions

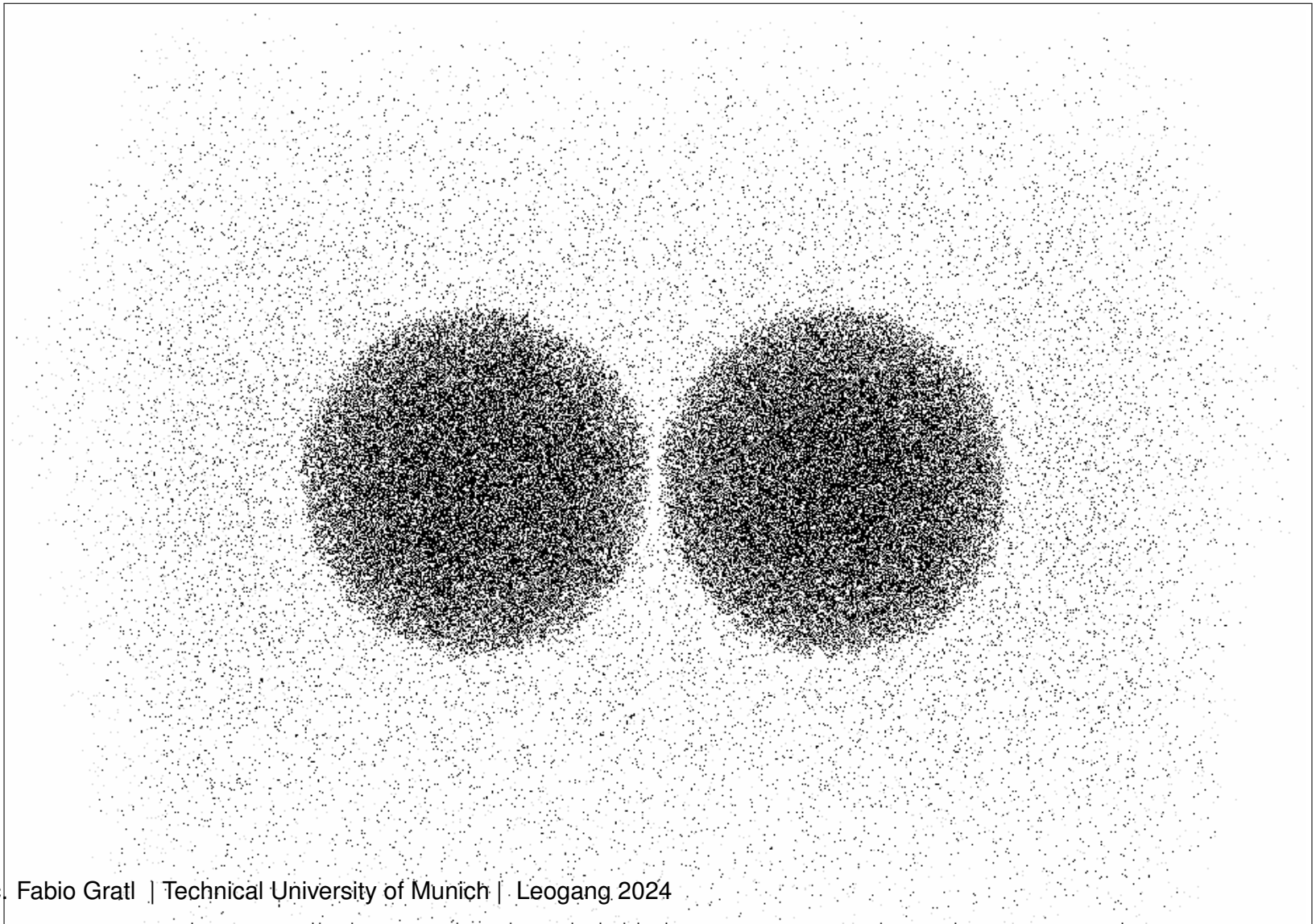


LArge-scale Deterministic Debris Simulation

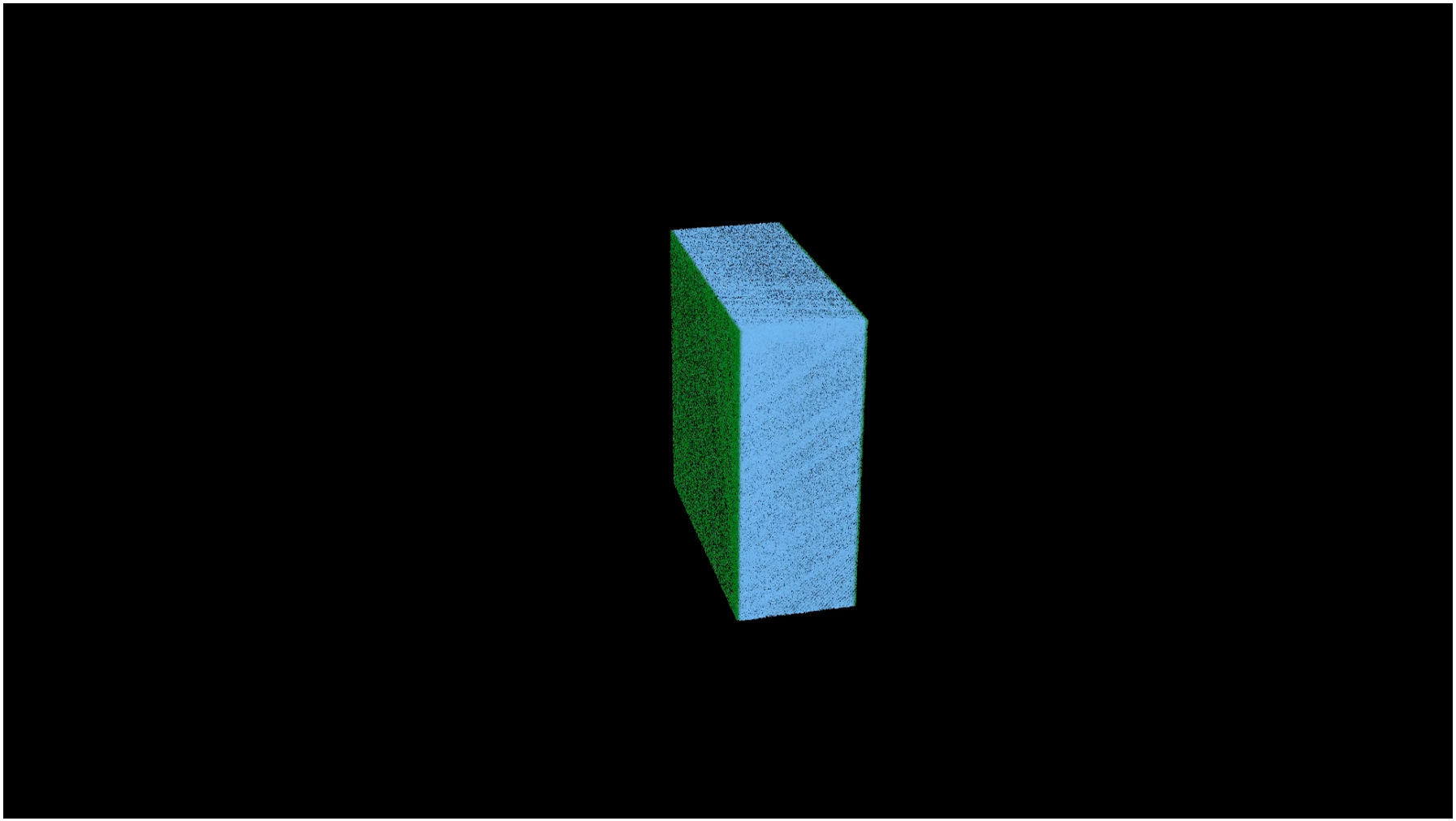
- Simulation of satellites as particles
- Trajectory modelling
- Newton's Laws of Motion
- Propagation perturbations
- Sub-timestep collision tracking
- N-Body problem $\Rightarrow O(N^2)$
- NASA Breakup Model
- https://github.com/esa/LADDS



Motivation: Molecular Dynamics Droplets



Motivation: Molecular Dynamics Exploding Liquid



Molecular Dynamics - Short Range

- Here: small rigid molecules as points without geometry
- Simulation of movement of particles
- Computation of pairwise forces
- Newton's Laws of Motion
- N-Body problem $\Rightarrow O(N^2)$
- Force cut-off $\Rightarrow O(N)$

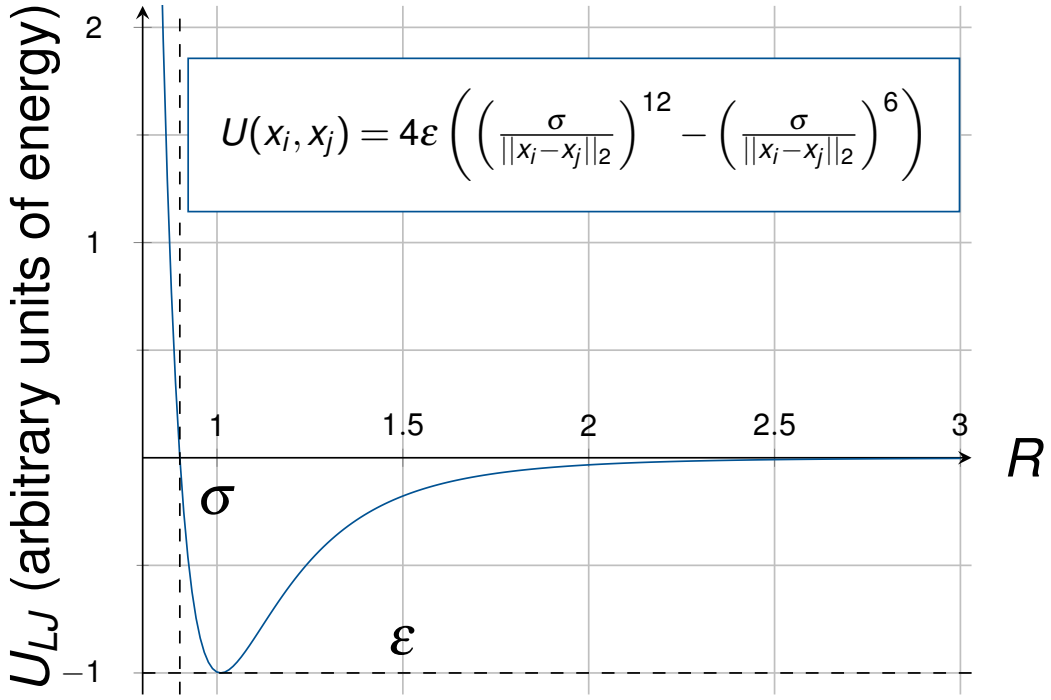
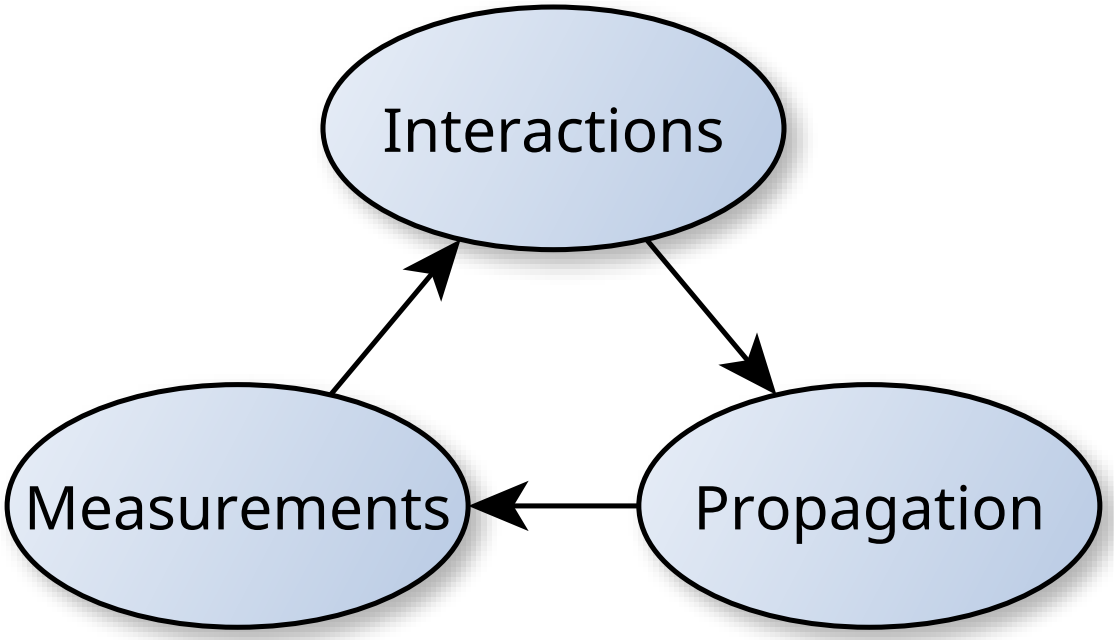


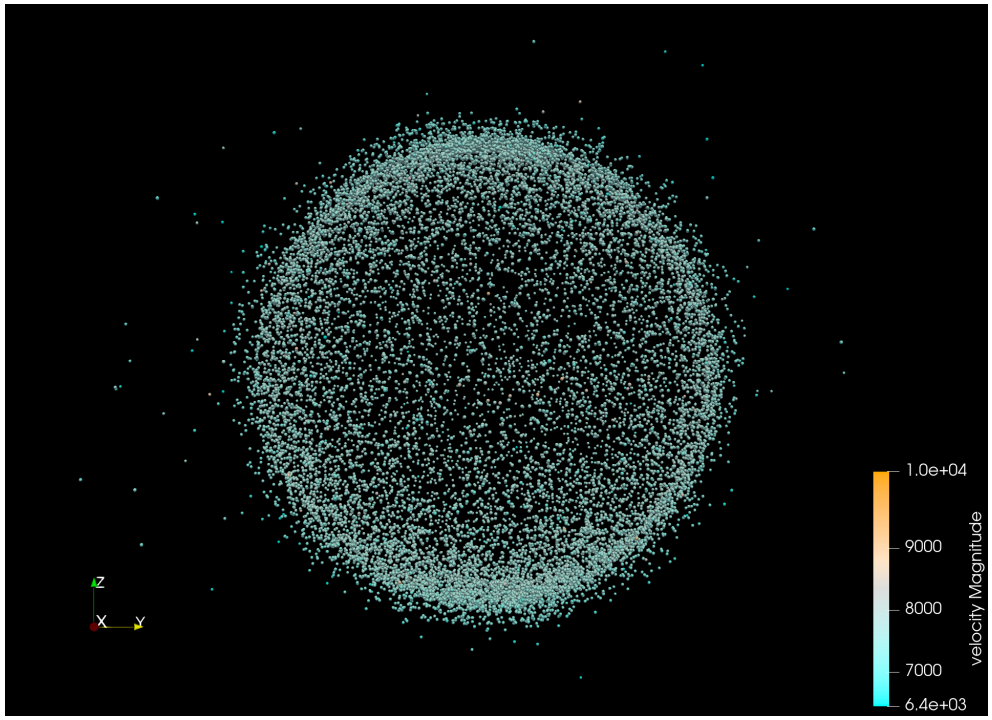
Figure: Lennard Jones Potential for $\varepsilon = 1$ and $\sigma = 0.9$

Particle Simulation Abstraction



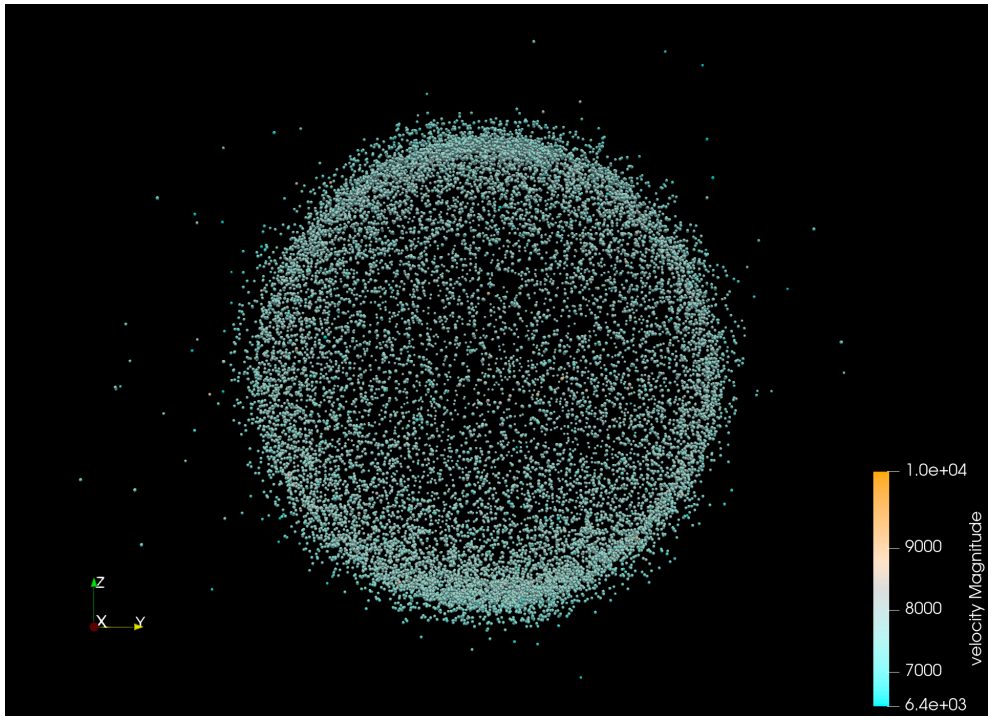
Particle Simulation Challenges

- Total number of particles
- Particle density
- (In-)Homogeneity
- Systems changing over time
- Many possible algorithms
- Arbitrary simulation types:
  MD, SPH, DEM, ...
- Overall goal:
  Minimize time to solution!



Particle Simulation Challenges

- Total number of particles
- Particle density
- (In-)Homogeneity
- Systems changing over time
- Many possible algorithms
- Arbitrary simulation types:
  MD, SPH, DEM, ...
- Overall goal:
  Minimize time to solution!
  Or energy!

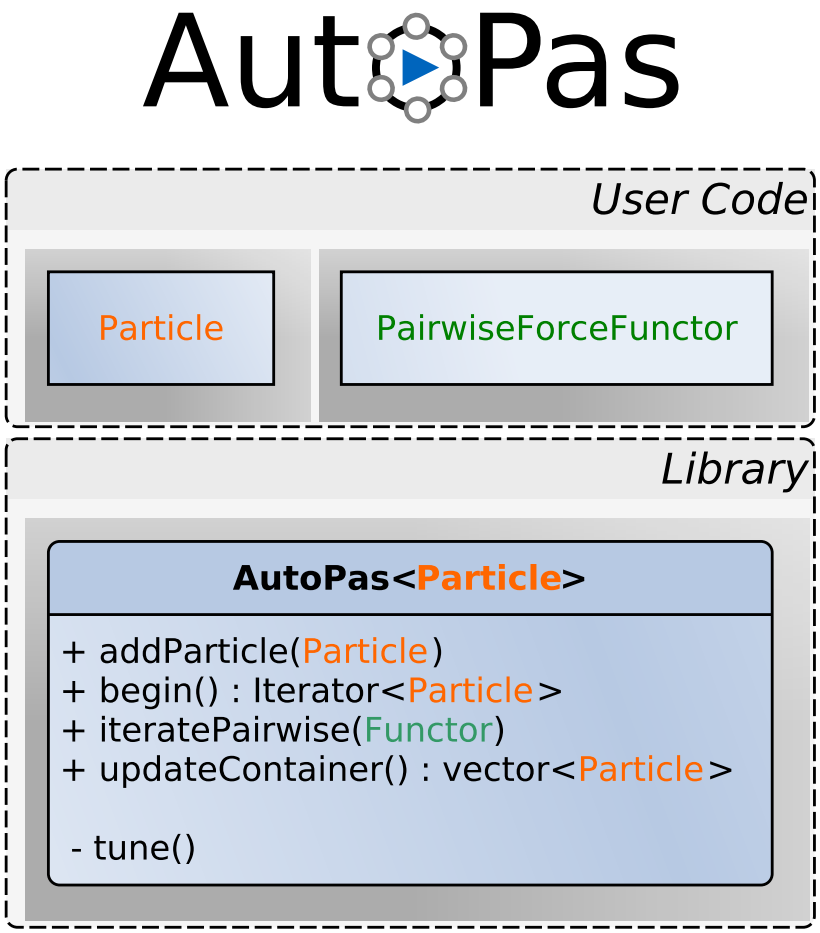


Introducing AutoPas

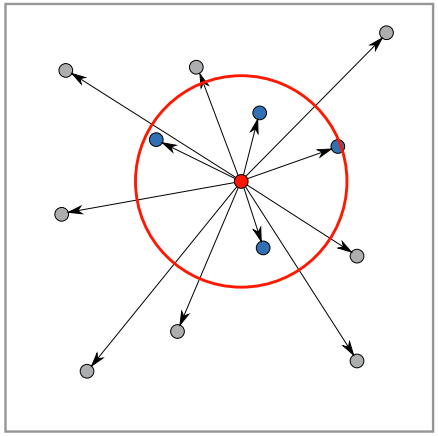
- Node-Level `C++17` library
- Black-box particle container
- Facade-like software pattern
- User defines:
  - Properties of particles
  - Force for pairwise interaction
- AutoPas provides
  - Containers, Traversals, Data Layouts, ...
  - Dynamic Tuning at run-time

⇒ General base for N-Body simulations

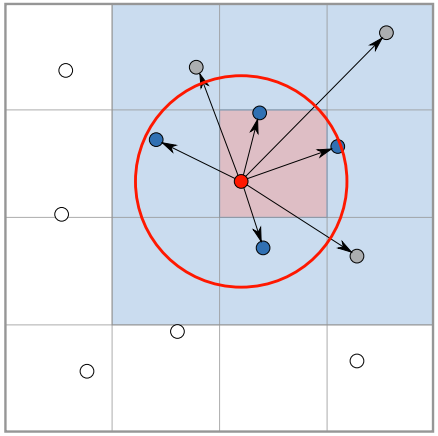
https://autopas.github.io/



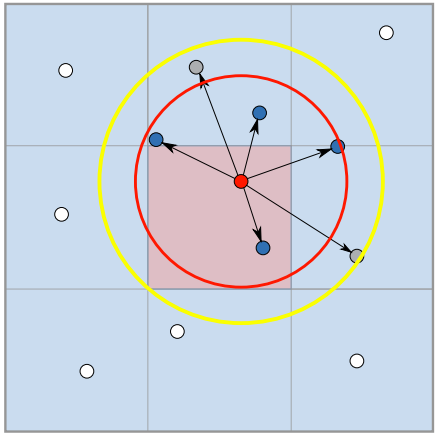
Algorithmic Choices



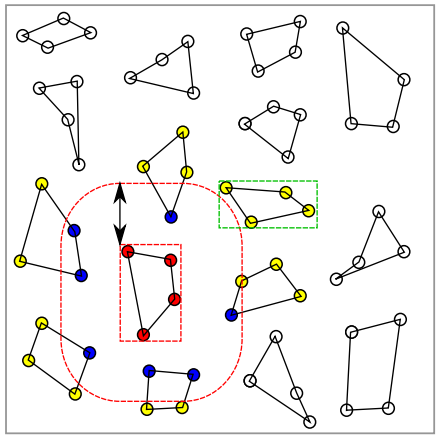
Direct Sum



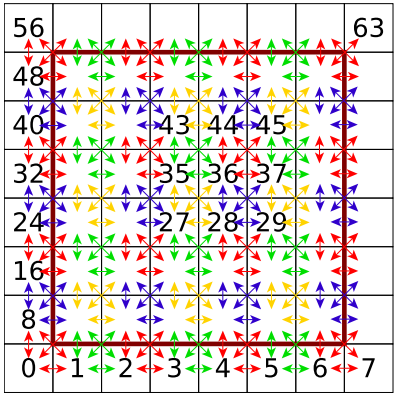
Linked Cells



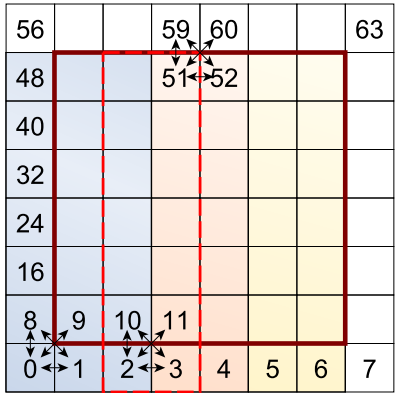
Verlet Lists



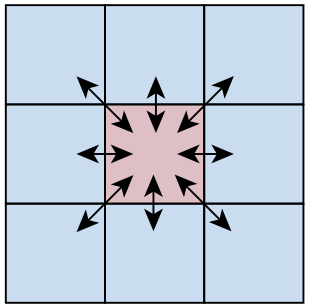
Verlet Cluster Lists



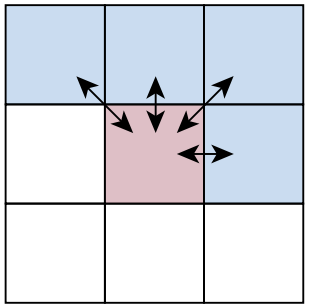
Coloring



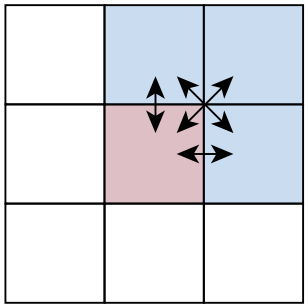
Slicing



C01



C18

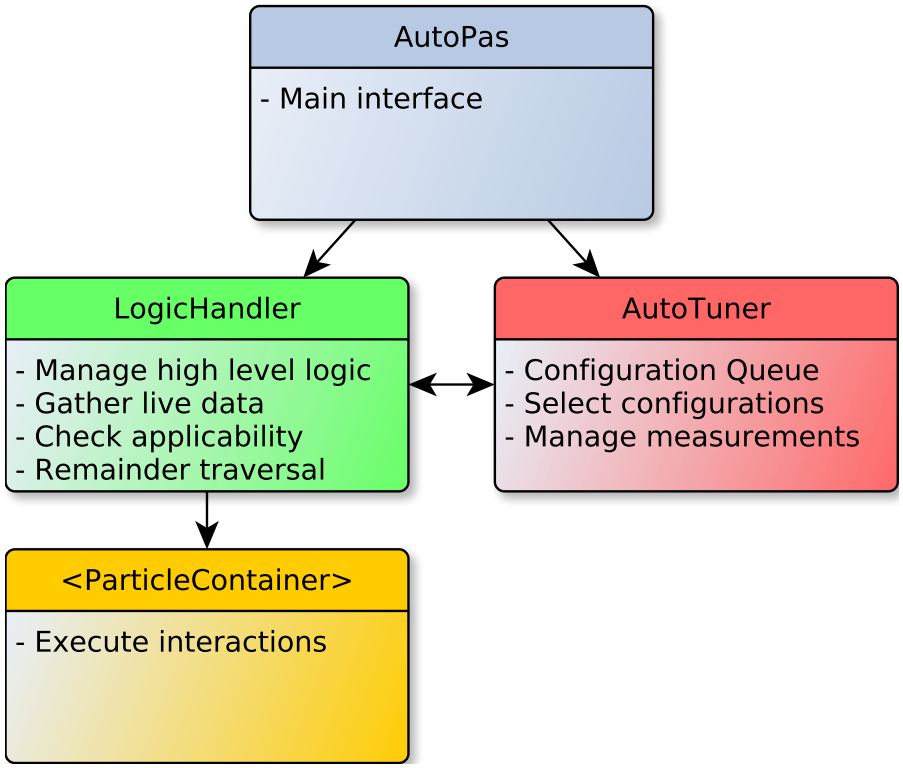


C08

Stencils

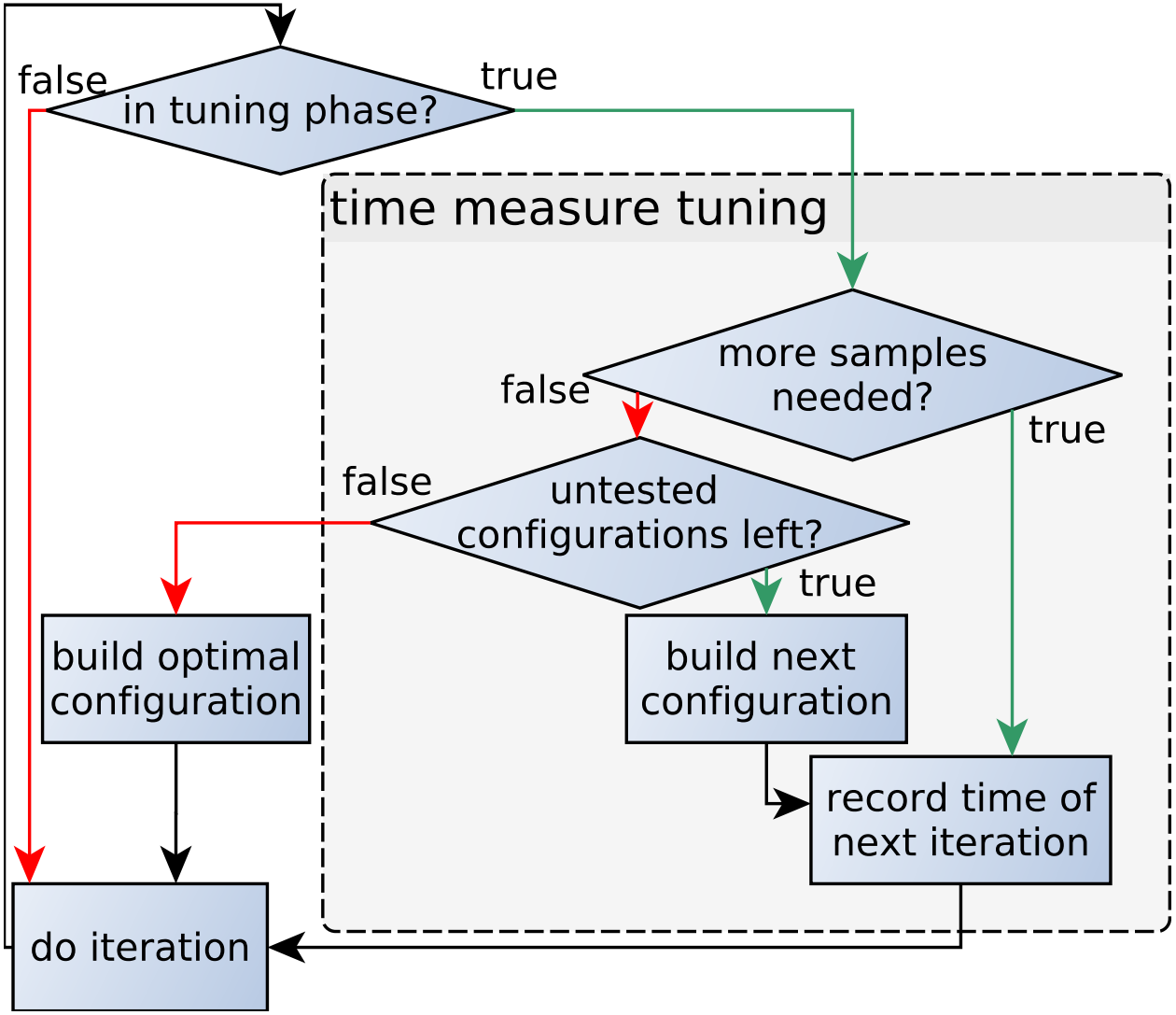
AutoPas Tuning Workflow

- Simple user interface
- Abstraction layer for non-critical common functionality
- Highly specialized data containers for particle interactions



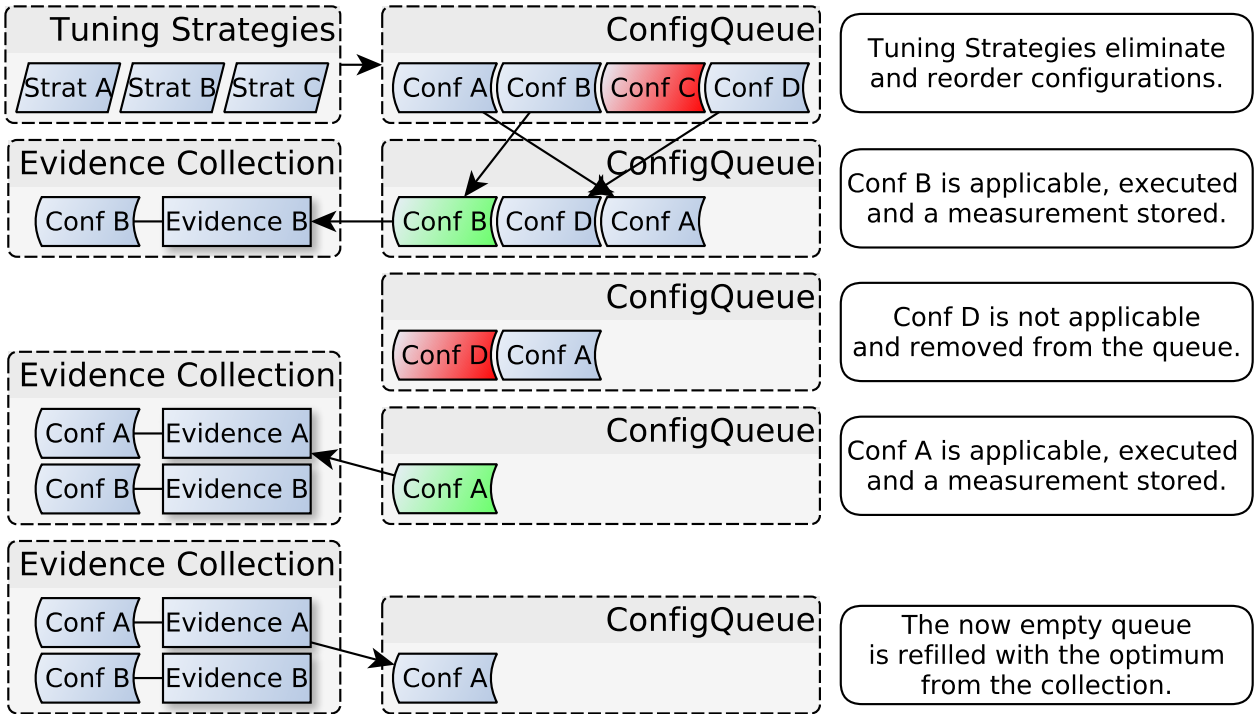
Tuning Cycle

- Common interfaces for containers, traversals, etc.
- Repeated periodically
- User can restrict search space

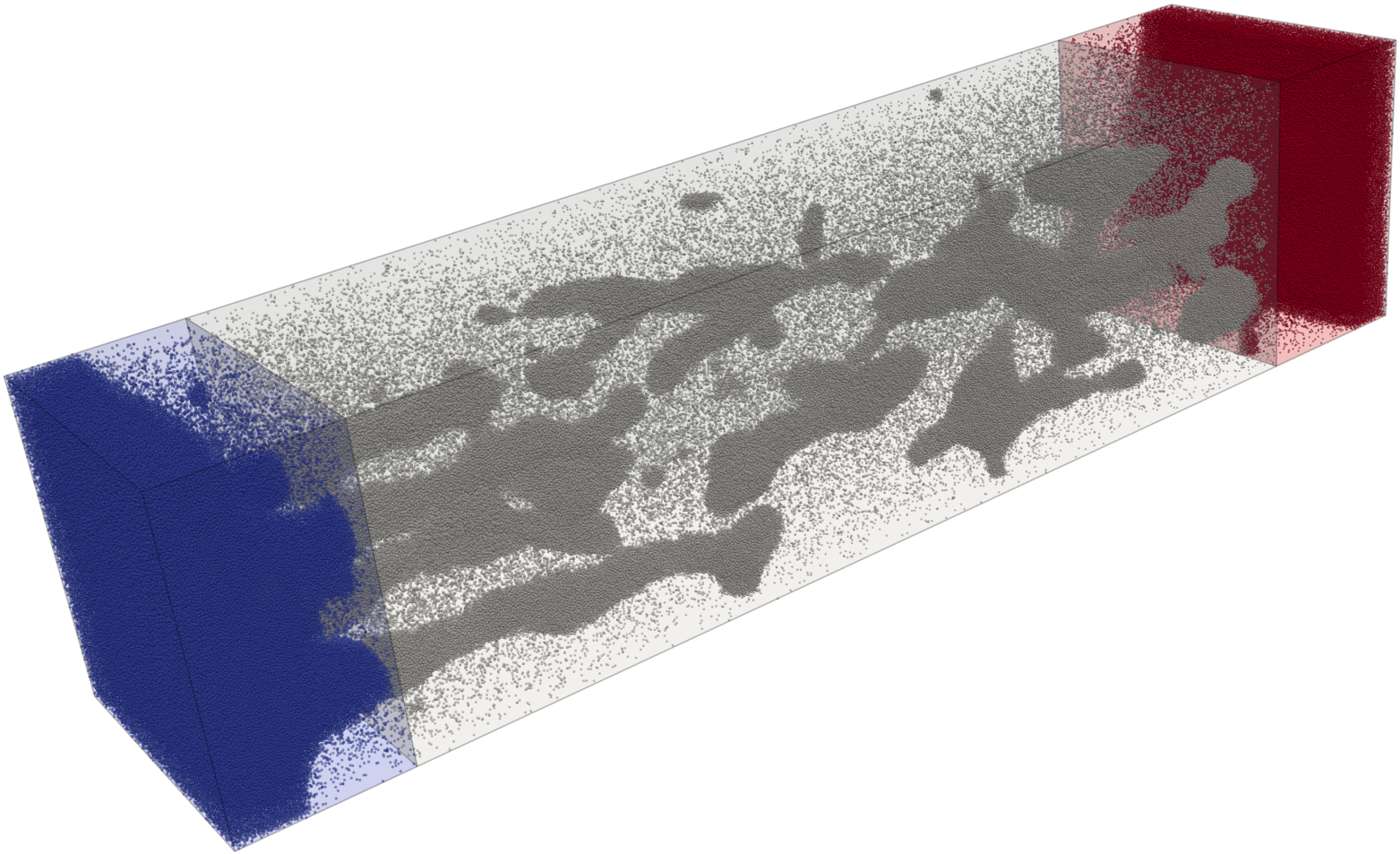


AutoPas Tuning Pipeline

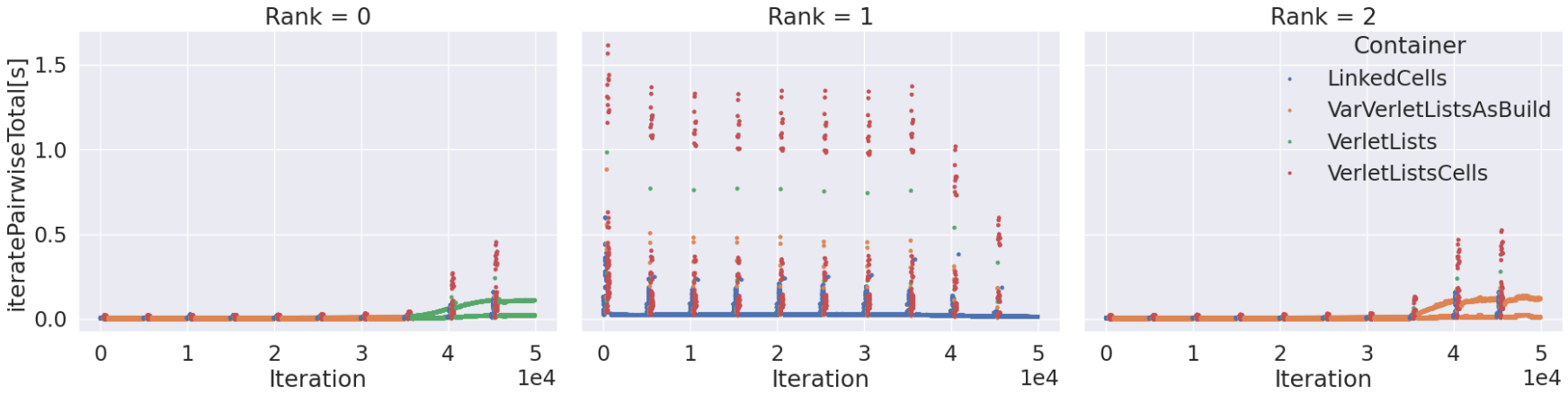
- Modular design
- Easily extensible with new:
  - ⇒ Configurations
  - ⇒ Tuning strategies
  - ⇒ Tuning metrics



ls1 mardyn + AutoPas $\Rightarrow$ MPI + Tuning



Iteration Performance over Time - Full Search



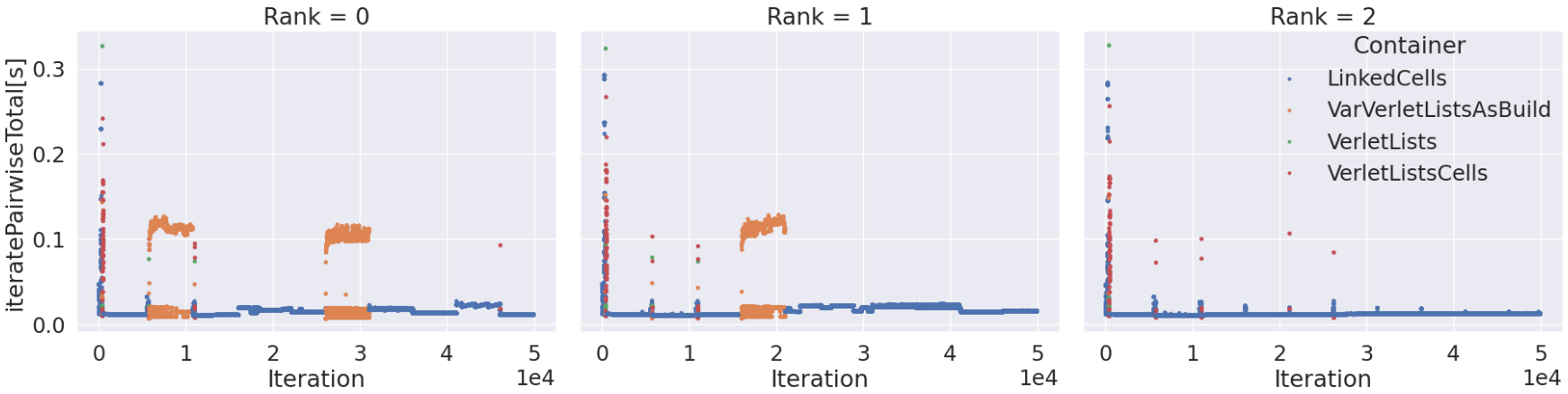
- Only first 50k iterations of 280k
- Exhaustive search through 92 configurations
- Tuning phases become extremely expensive
- Highest outliers are Verlet list rebuilds

Rule Based Tuning

```
if numParticles > 1000:
  [container="LinkedCells"] >= [container="DirectSum"];
endif
```

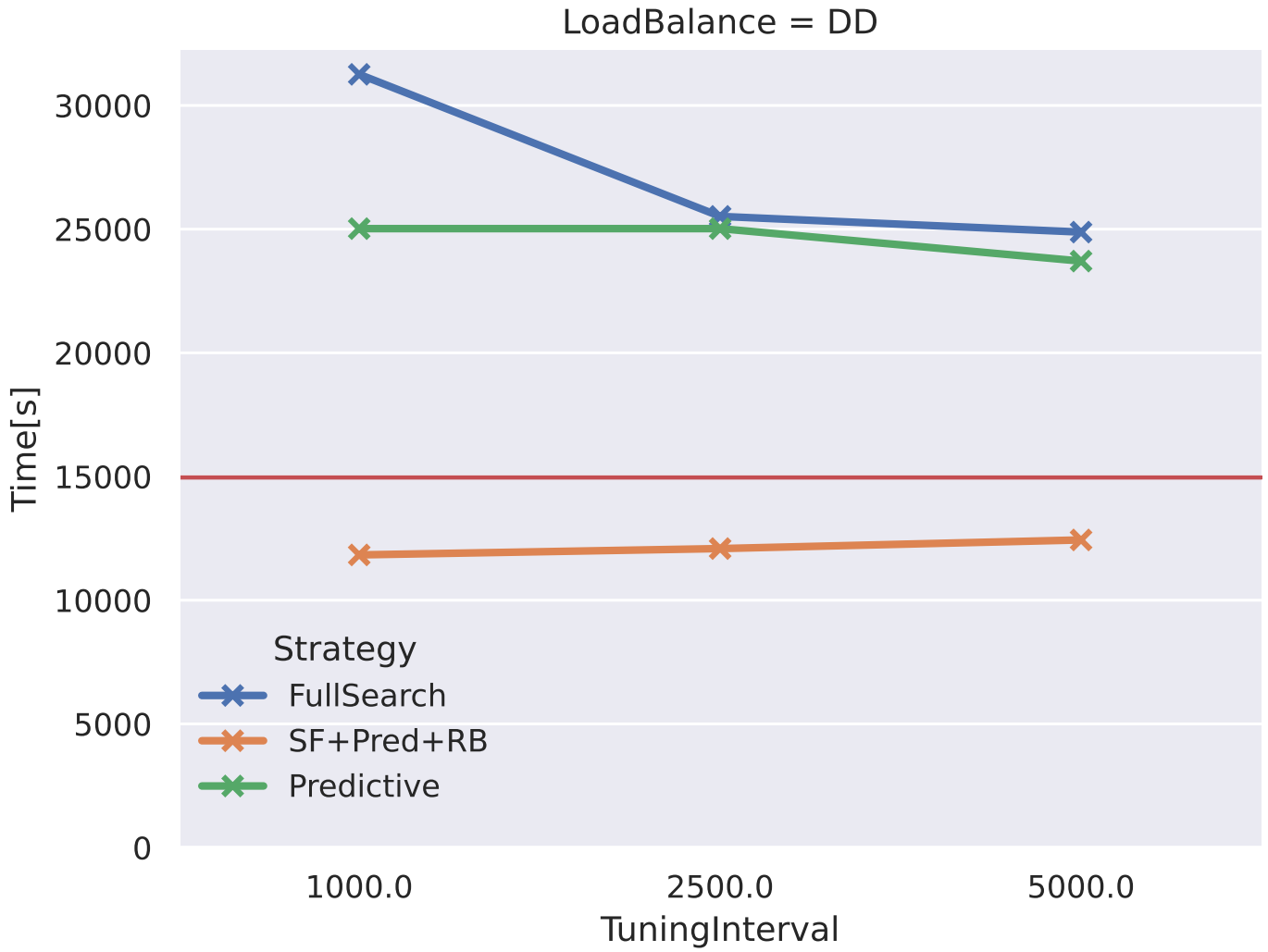
- Domain Specific Language
- Filters based on pattern matching
- Can make use of dynamic live data
- Implemented via virtual machine

Iteration Performance over Time - Smart



- Y-axis 5x shorter!
- Apply expert rules to avoid inefficient configurations from the start
- Apply learning to only test promising configurations

Smart Tuning Outperforms Everything



Summary

- AutoPas enables automated algorithm selection on node level.
  - Independent tuning of all MPI ranks.
  - Optimal algorithm choices also for non-expert users.
- Black Box particle container facilitates development of new applications.
  - Adapts to changing requirements.
  - No software expert knowledge needed.
- Speed up automated algorithm selection by combination of knowledge and learning.