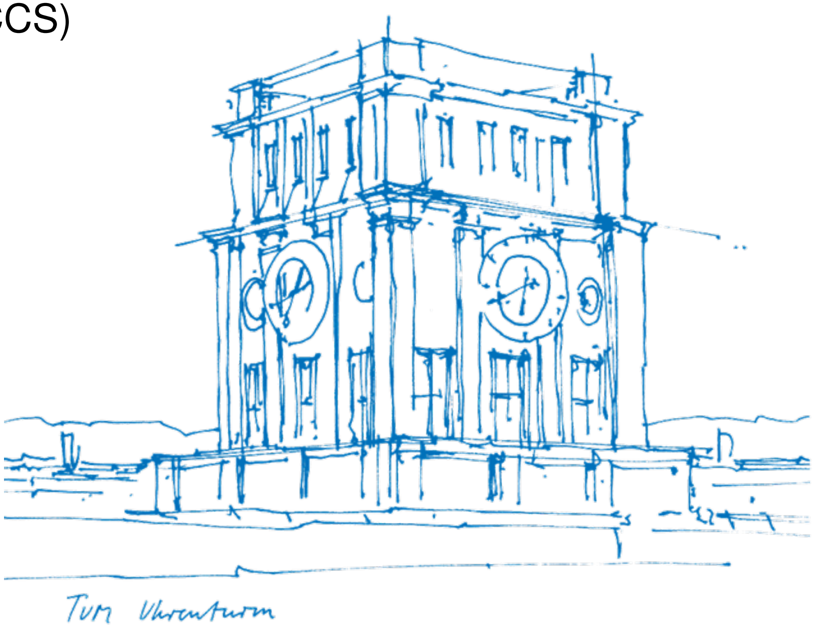
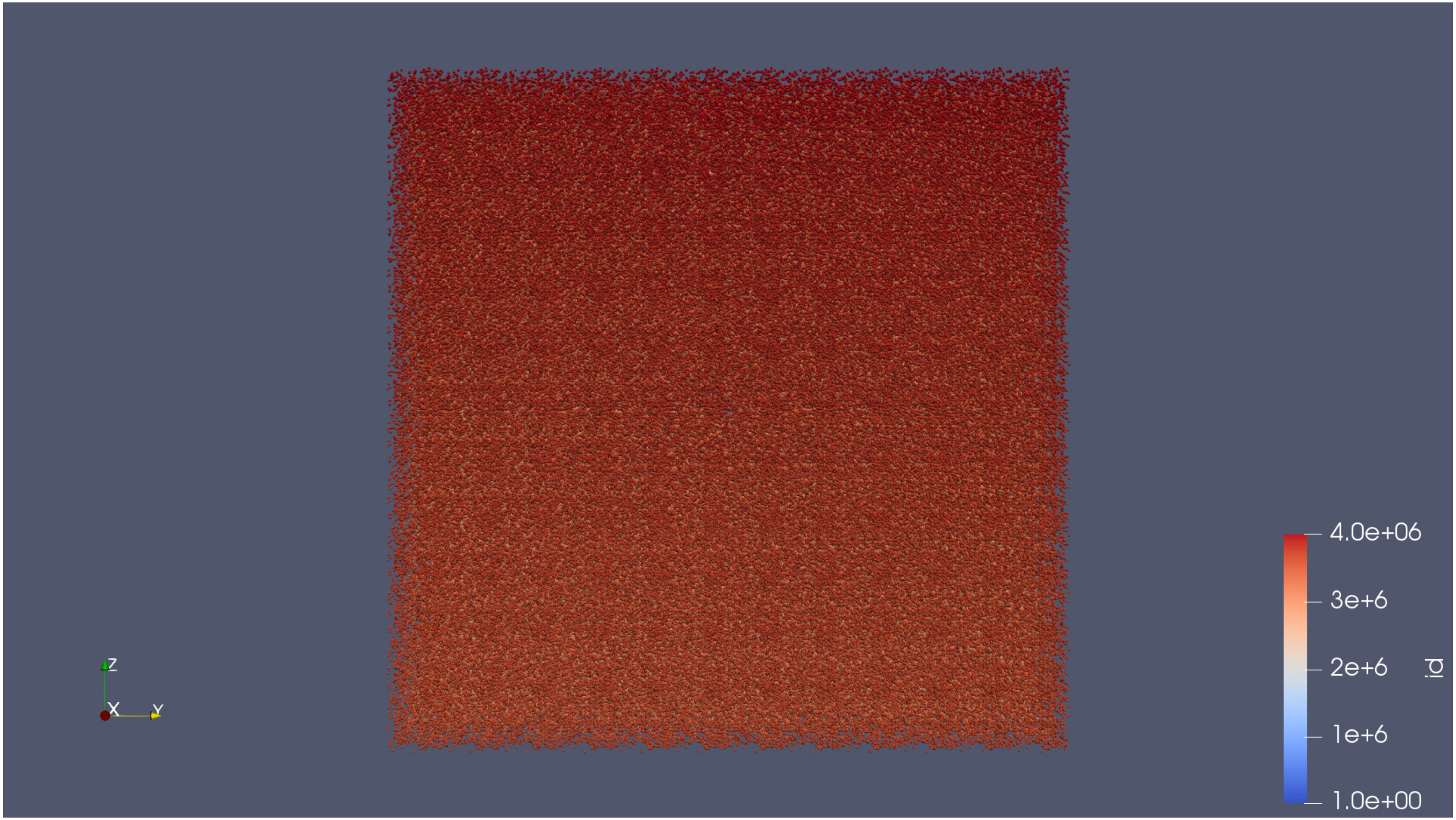
Leveraging Node-Level Performance for Molecular Dynamics through Auto-Tuning

Fabio Gratl
Technical University of Munich
Faculty of Informatics
Chair of Scientific Computing in Computer Science (SCCS)
Spokane, February 28. 2019



Motivation

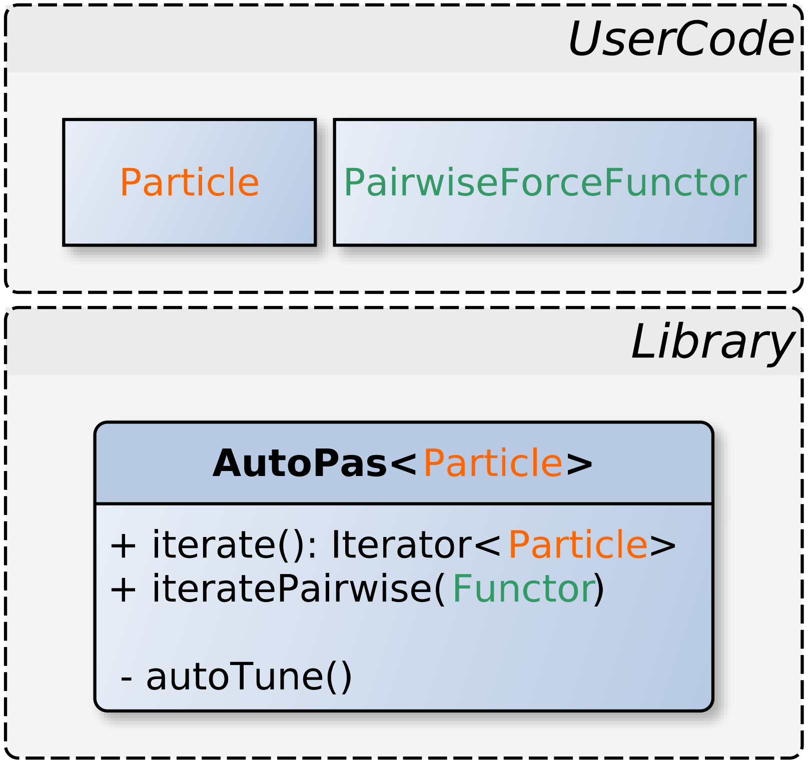


AutoPas

AutoPas: Overview

- Node-Level C++ header library
- User defines:
  - Properties of particles
  - Force for pairwise interaction
- AutoPas provides:
  - Containers, Traversals, Data Layouts, ...
  - Dynamic Tuning at run-time
- Black Box container

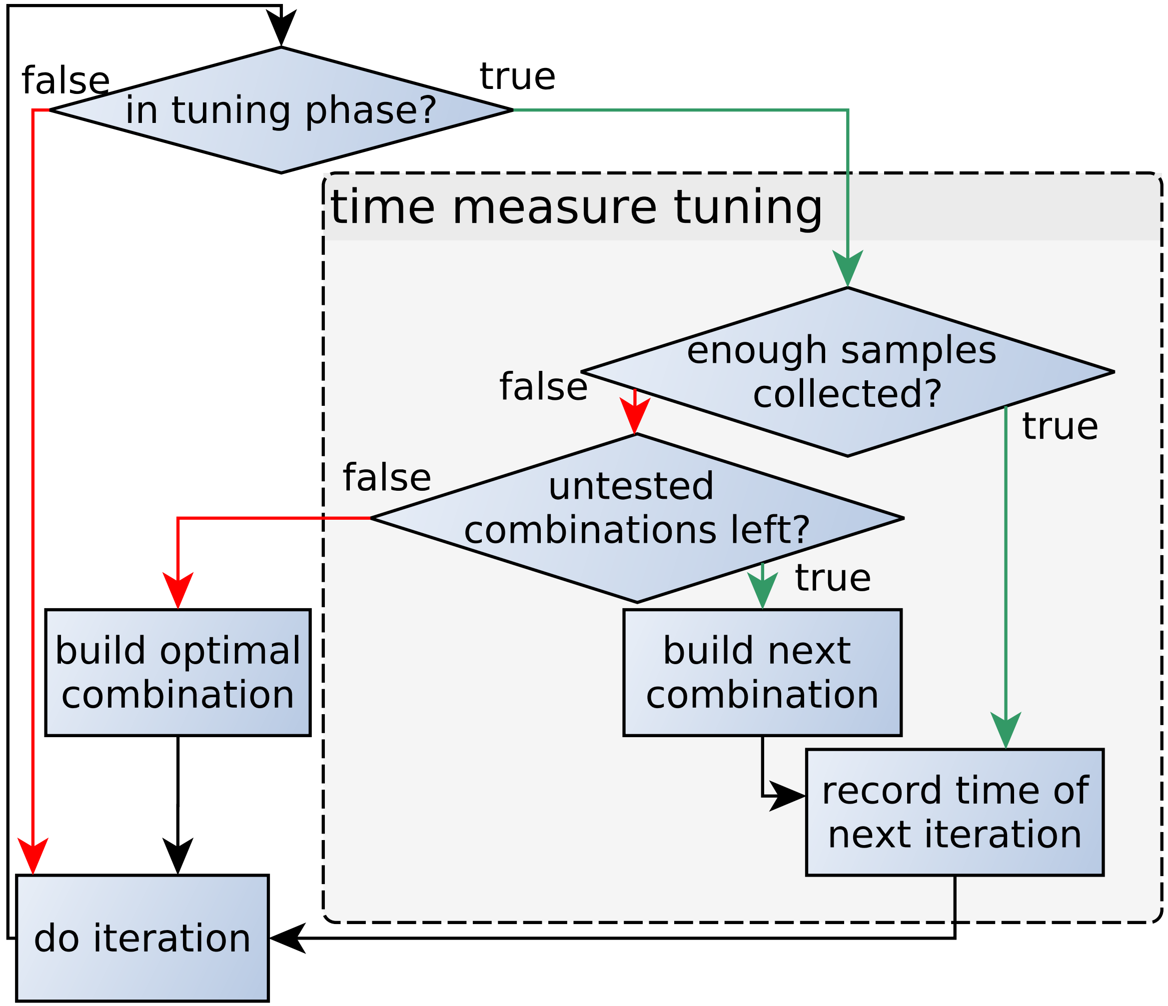
⇒ General base for *N*-Body simulations



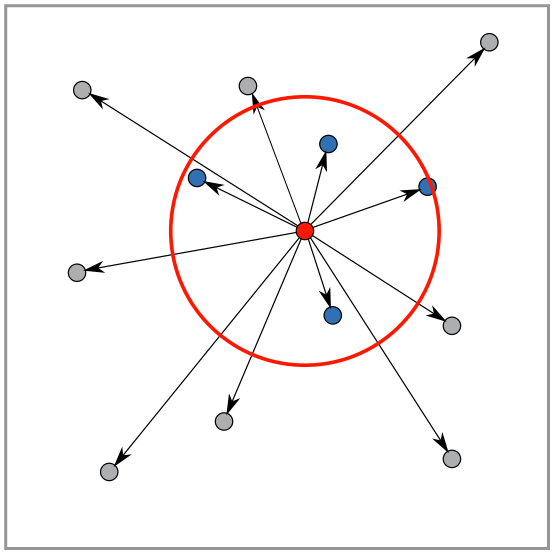
https://github.com/AutoPas/AutoPas

Auto-Tuning Process

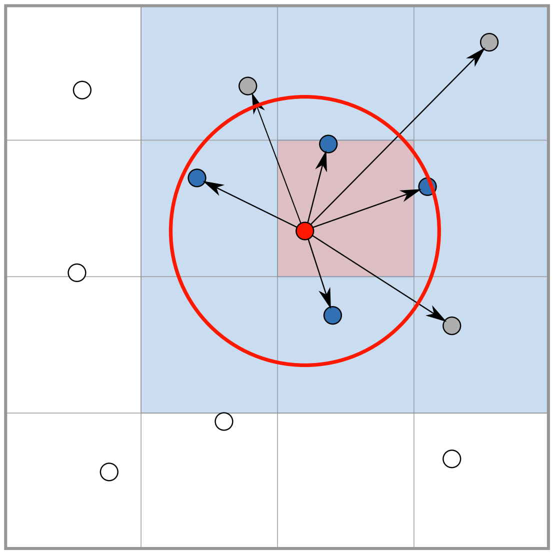
- Common interfaces for containers, traversals, etc
  $\Rightarrow$ Strategy pattern
- Repeated periodically
- User can restrict testing space



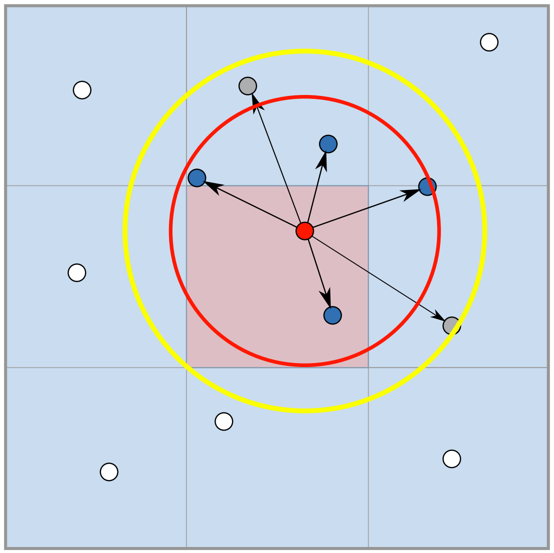
Container Options



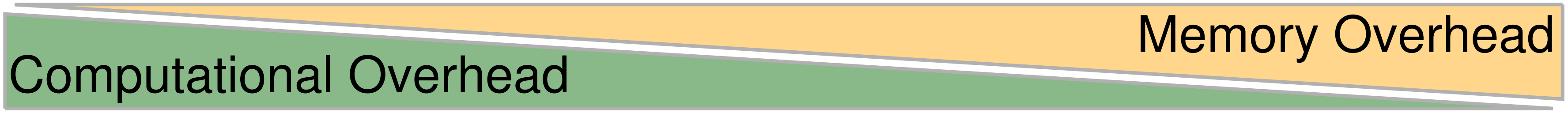
Direct Sum



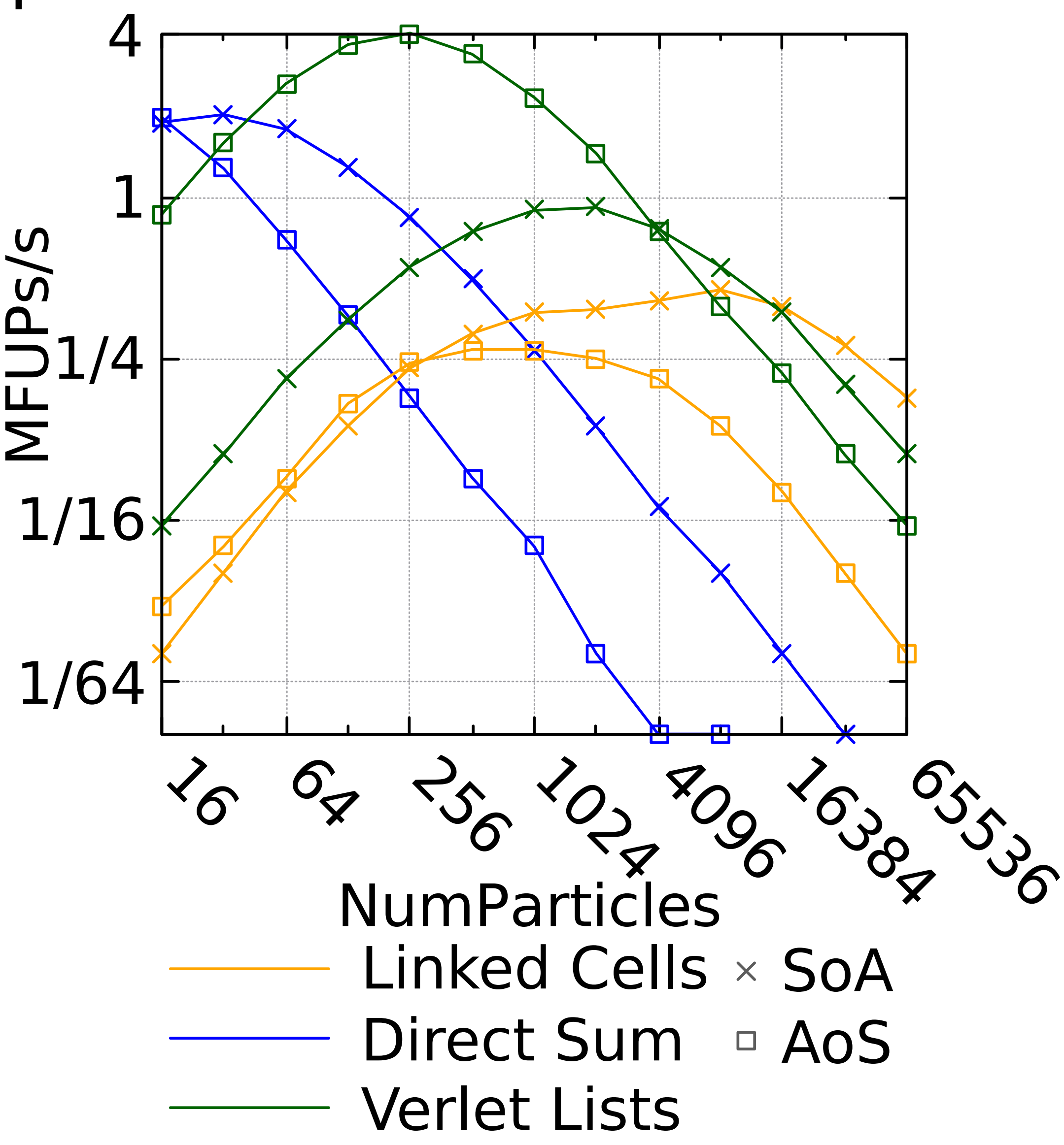
Linked Cells



Verlet Lists

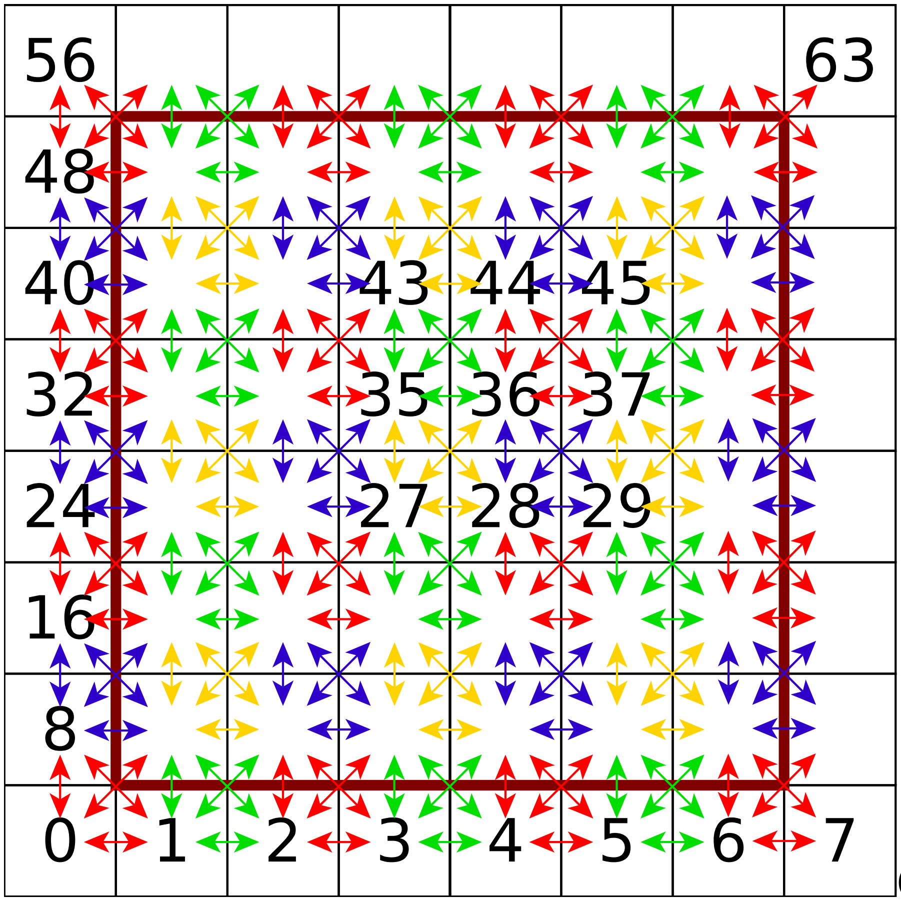


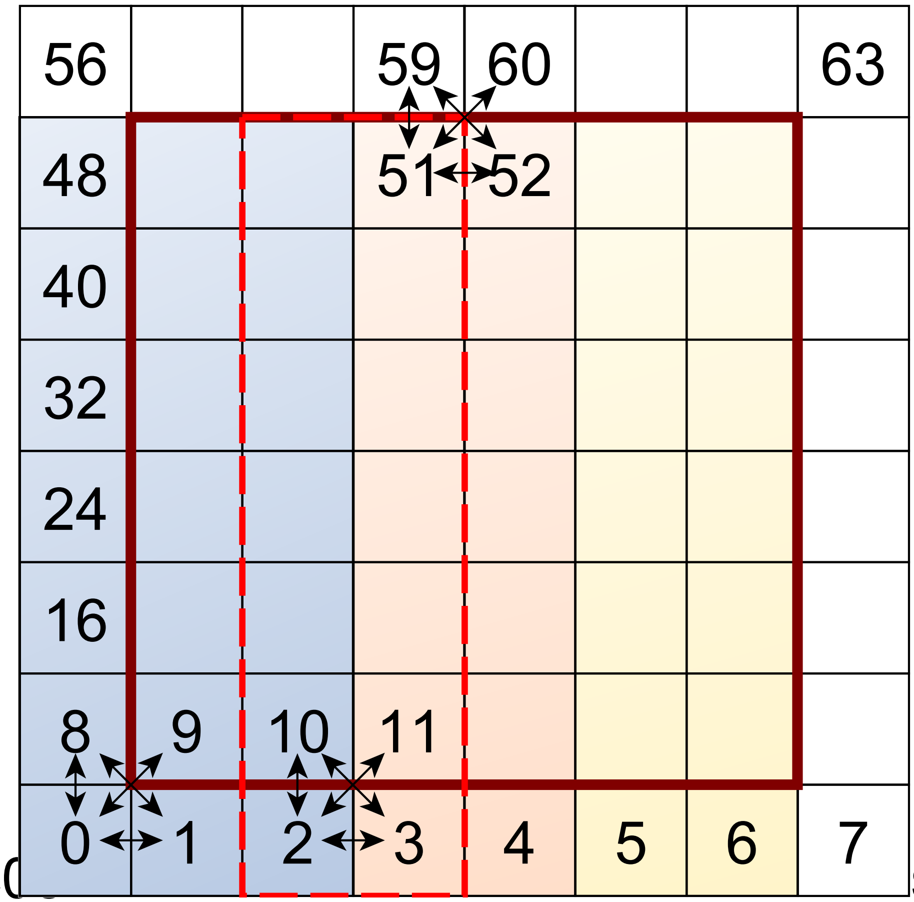
Container Comparison



- Every container has advantages
- Linked Cell benefits most from SoA
  $\Rightarrow$best in dense scenarios

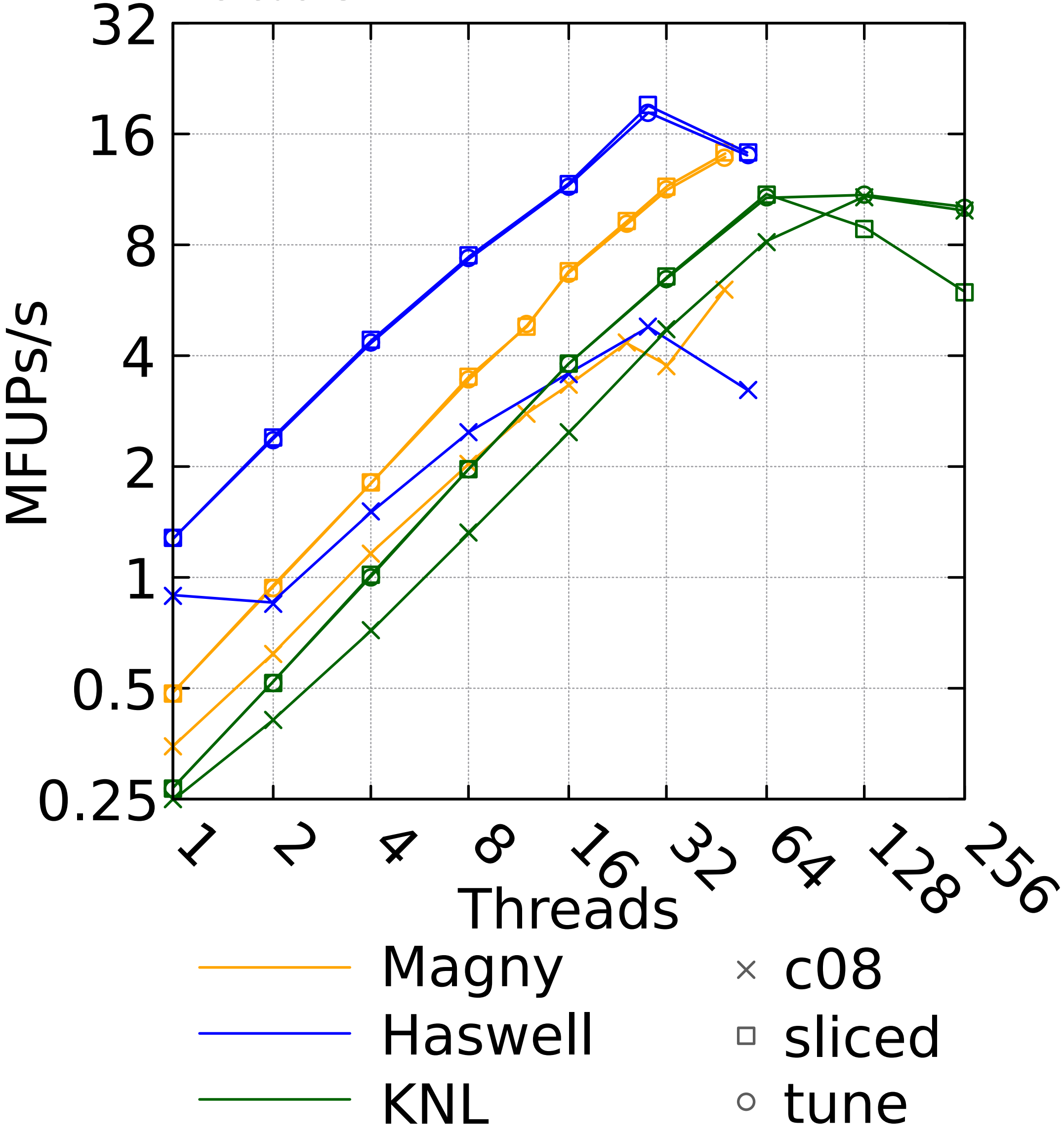
Linked Cells Parallelization Options





Hardware Comparison / Threads

- Traversals:
  c08: 8-way domain coloring
  sliced: regular 1D domain partitioning

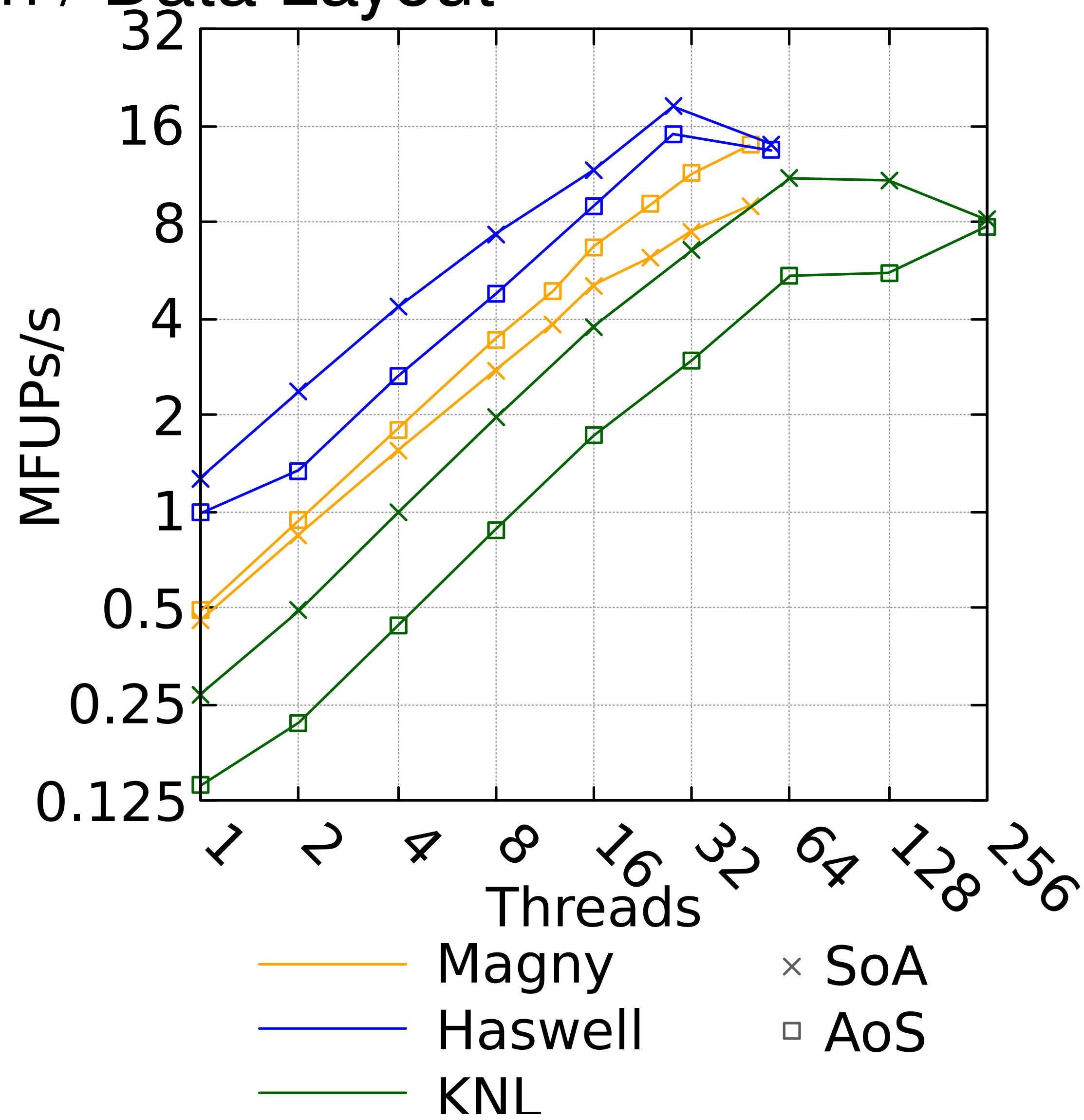


Hardware Comparison / Data Layout

- Vector Instructions:

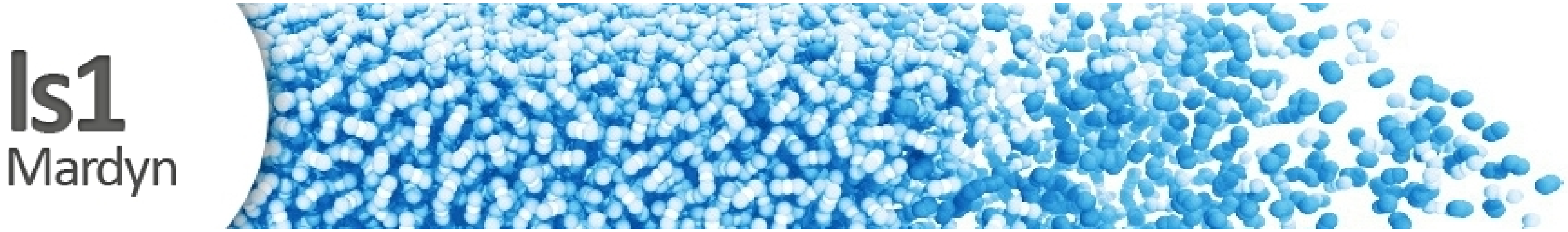
Magny	SSE4	128
Haswell	AVX2	256
KNL	AVX512	512

⇒ Optimal data layout dependent on hardware



Integration into ls1 mardyn

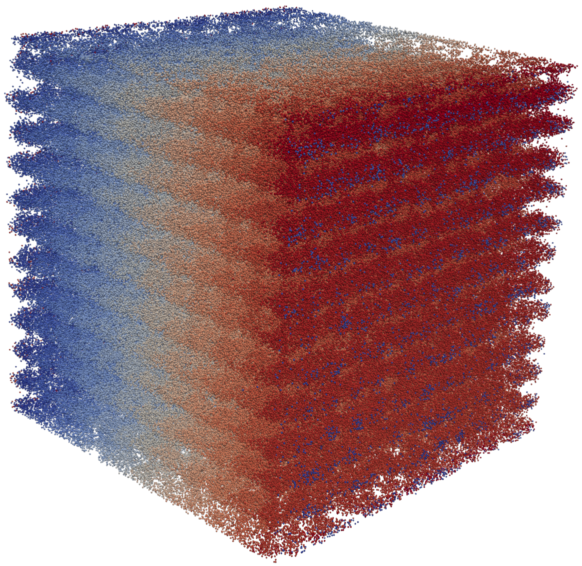
- ls1 mardyn:
  - Large number of small rigid molecules.
  - Actively used in chemical engineering.
- Example Lennard-Jones functor from AutoPas
- New particle class
  - Inherits from AutoPas and ls1 mardyn particle interface.
  - Acts as coupler
- New particle container class
  - Only wrapper around AutoPas main interface.

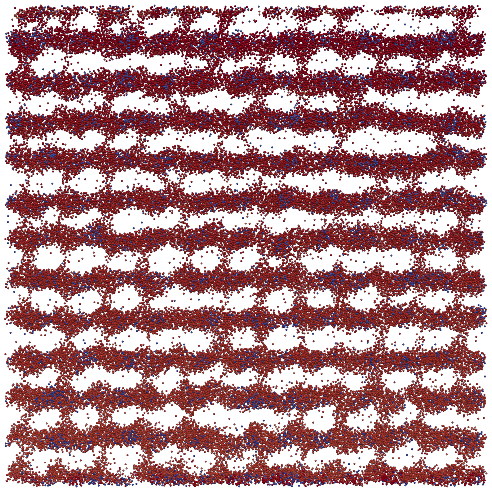


Scenario: Spinodal Decomposition

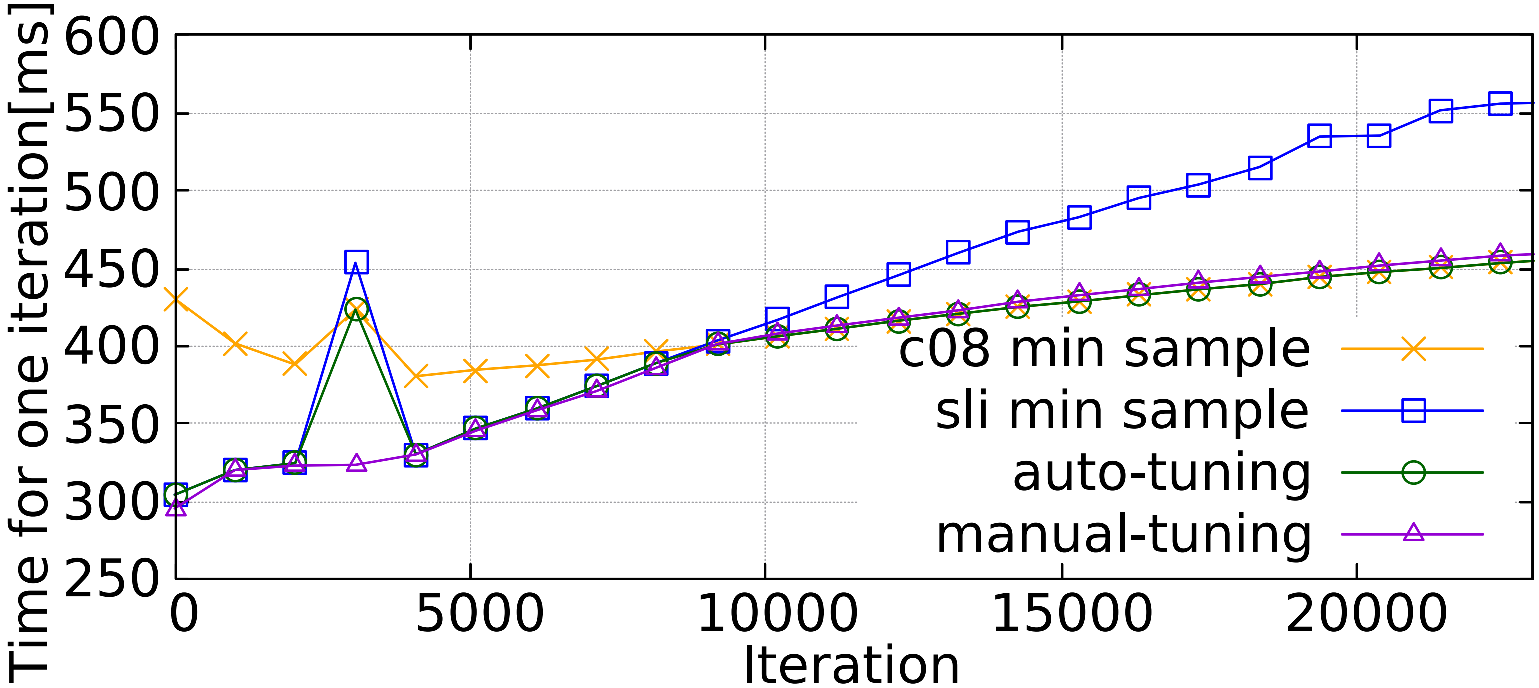
- 4 008 960 particles
- Periodic boundaries
- Sub-critical temperature
- Rapid and drastic change in homogeneity

  ⇒ Interesting target for tuning!



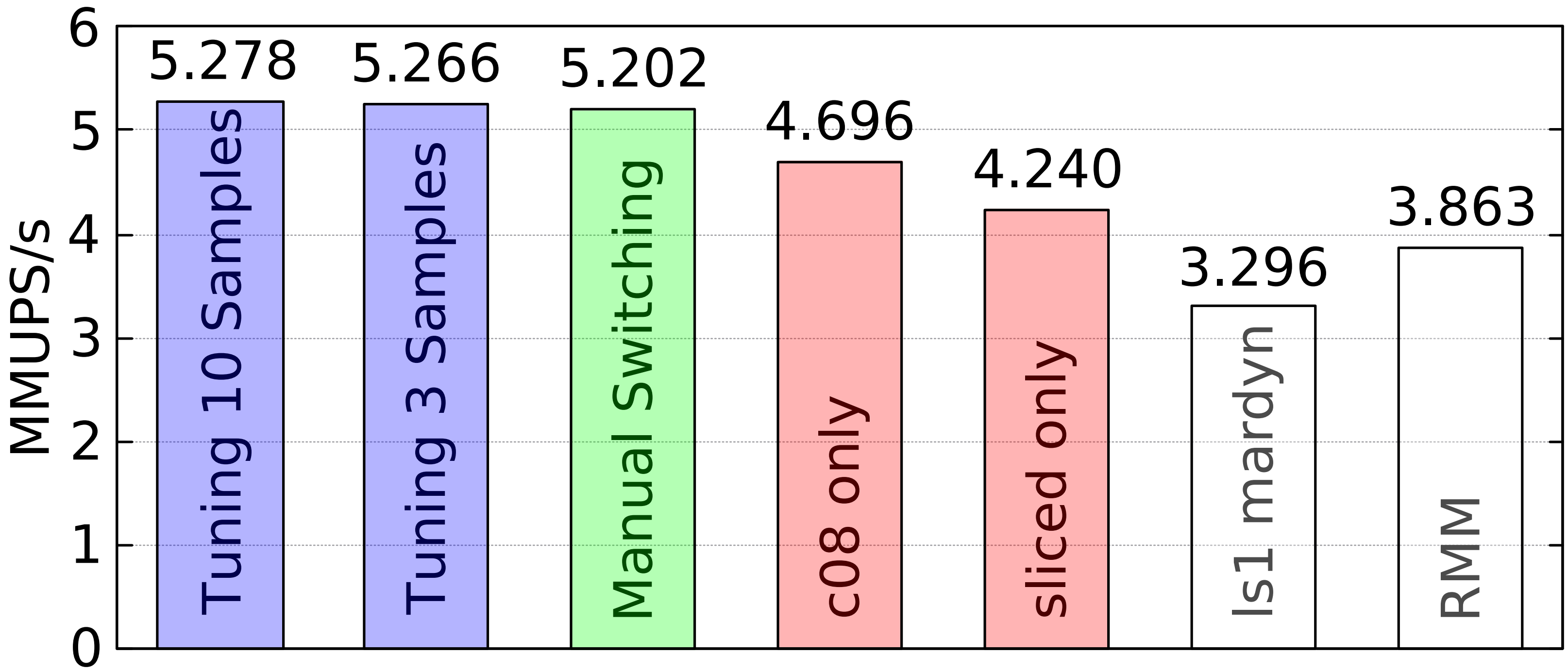


Tuning Behavior



- Tuning switches as expected
- Misclassification can happen

Tuning Overhead



- Tuning and manual switching equally fast.
  $\Rightarrow$ No overhead from tuning
- Tuning faster than static configuration.
- Faster than original ls1 mardyn, even in Reduced Memory Mode.

Conclusions

- AutoPas is a black box *N*-Body container.
- Dynamic tuning enables optimal performance for changing scenarios.
- Achievable for users without expert knowledge.
- Easy to integrate in existing codes.

... and future work:

- More algorithms.
- More tuning parameters.
- Search space reduction.