Optimizing Molecular Dynamics Simulations with Dynamic Auto-Tuning

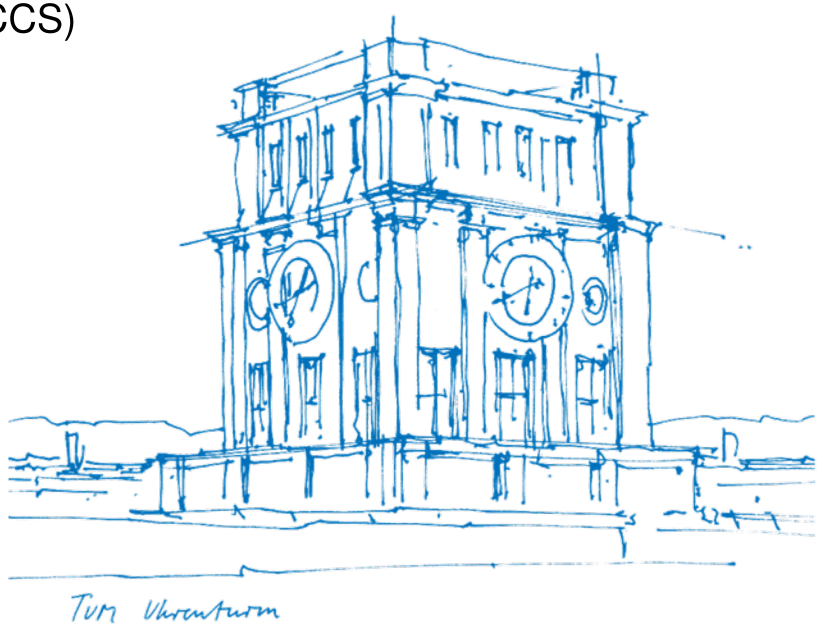
Fabio Gratl, Steffen Seckler, Hans-Joachim Bungartz, Philipp Neumann

Technical University of Munich

Department of Informatics

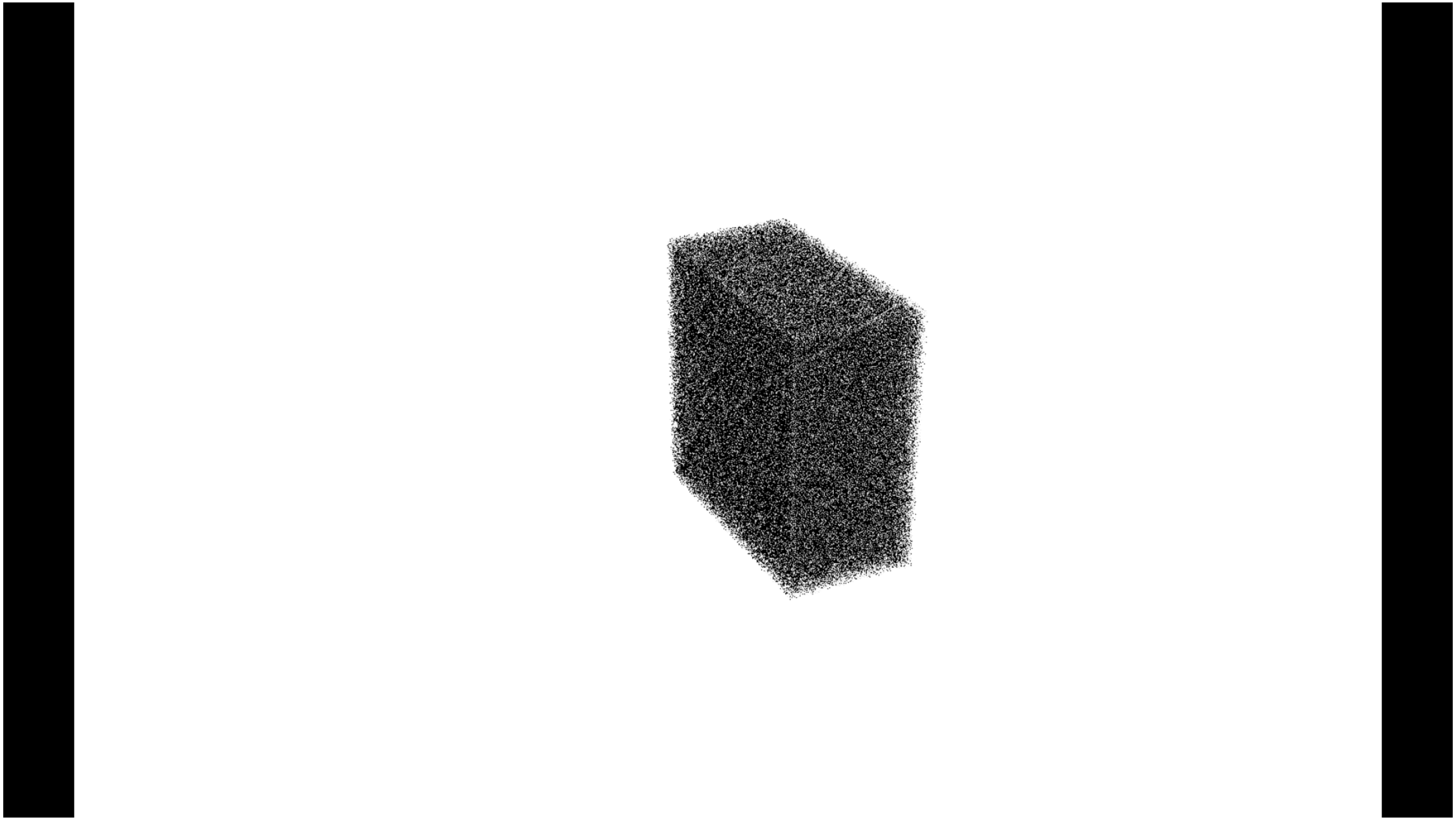
Chair of Scientific Computing in Computer Science (SCCS)

Seattle, February 13. 2020



Bundesministerium
für Bildung
und Forschung

Motivation



Molecular Dynamics

Molecular Dynamics - Short Range

- Here: small rigid molecules
- Simulation of movement of particles
- Computation of pairwise forces
- Newton's Laws of Motion
- N-Body problem $\Rightarrow O(N^2)$
- Force cut-off $\Rightarrow O(N)$

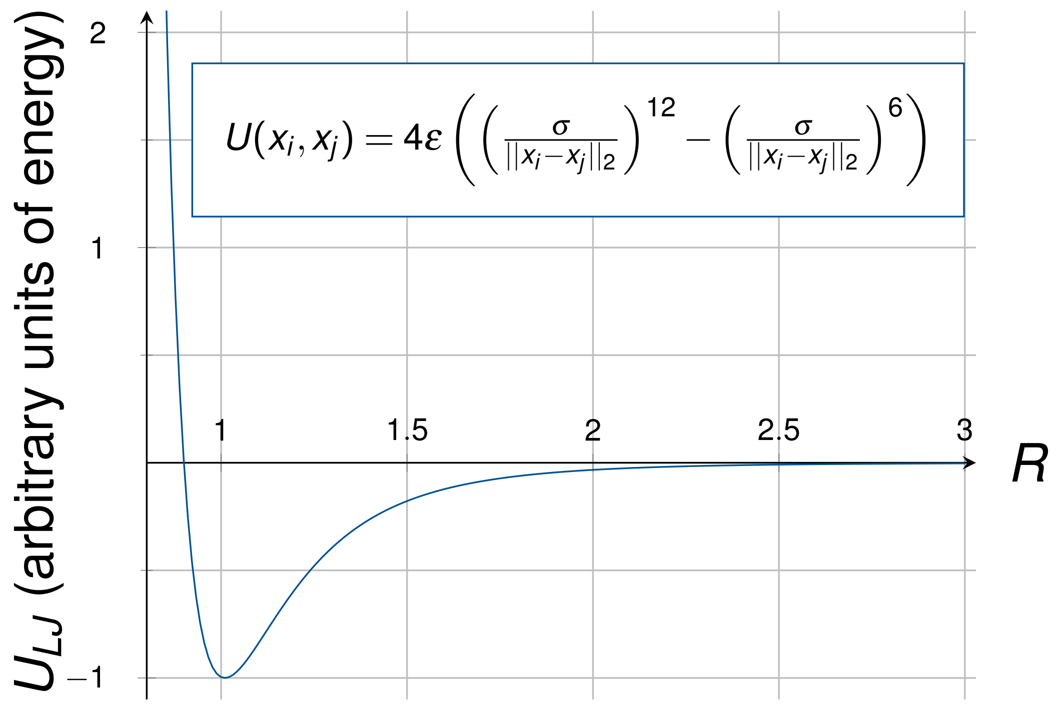
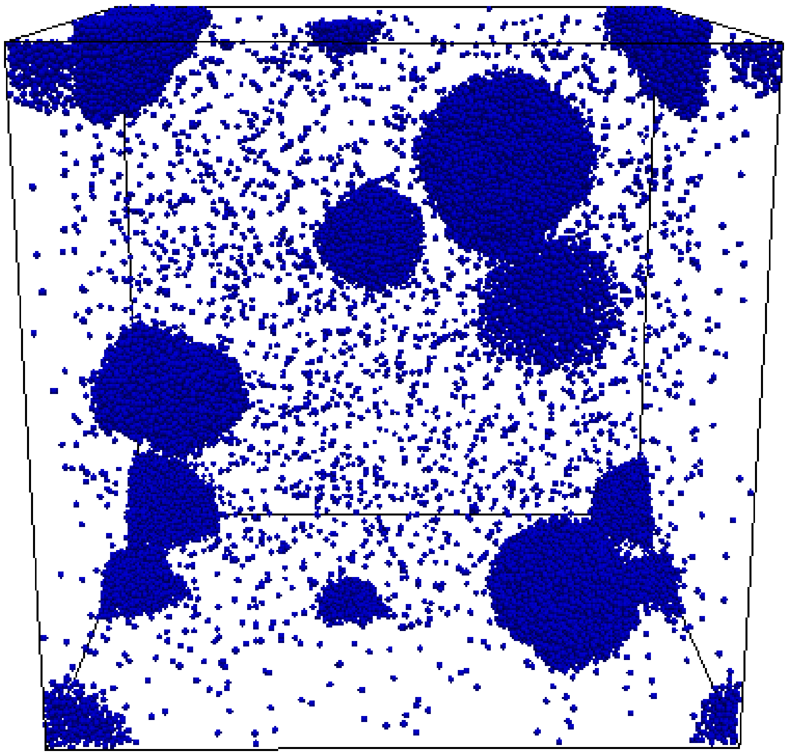


Figure: Lennard Jones Potential for $\varepsilon = 1$ and $\sigma = 0.9$

Challanges

- Total number of particles
- Particle density
- (In-)Homogeneity
- Systems changing over time
- Many possible algorithms
- Overall goal:
  Minimize time to solution!



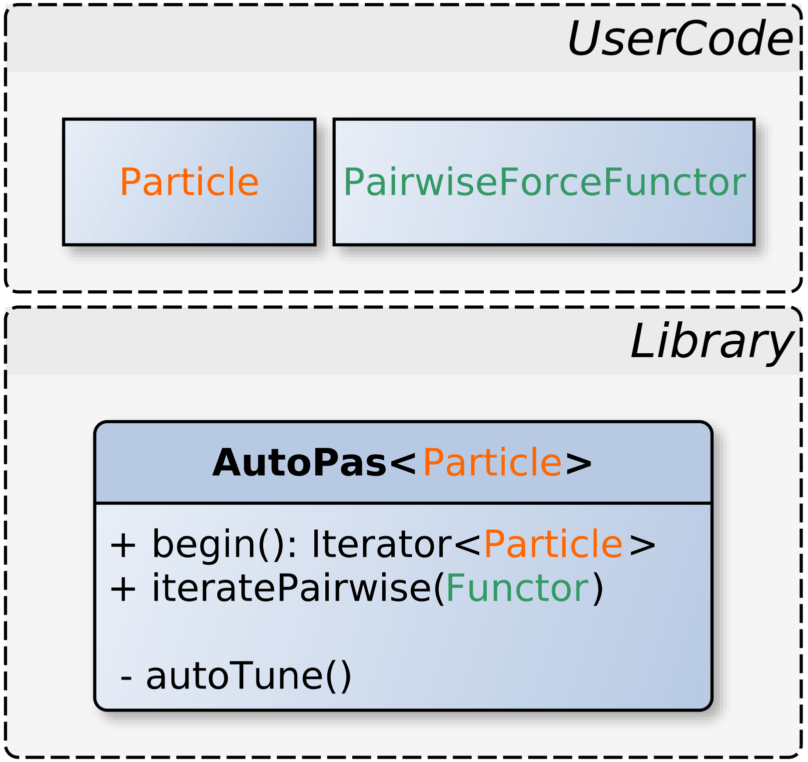
AutoPas

AutoPas: Overview

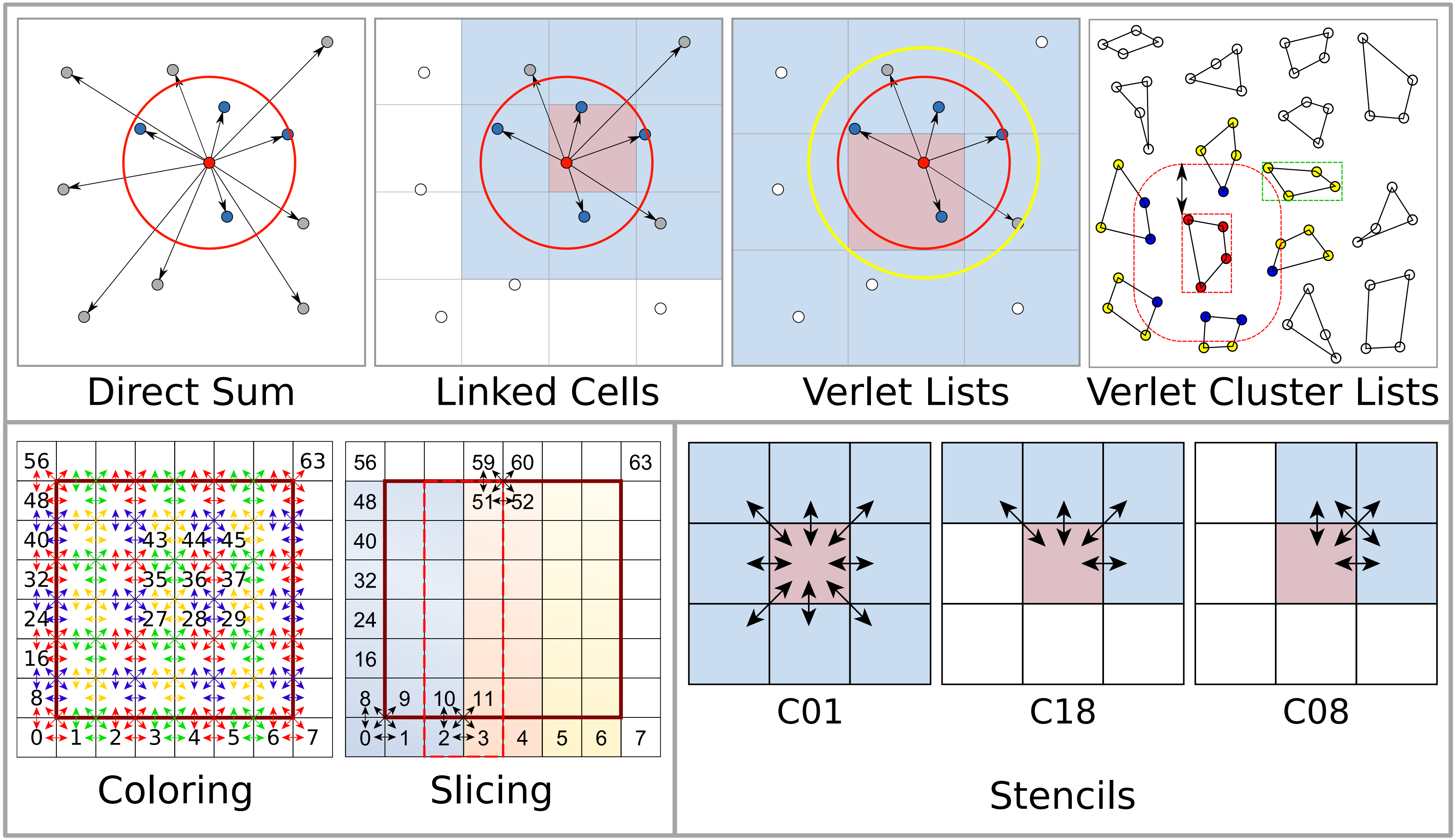
- Node-Level C++ library
- Black Box container
- User defines:
  - Properties of particles
  - Force for pairwise interaction
- AutoPas provides:
  - Containers, Traversals, Data Layouts, ...
  - Dynamic Tuning at run-time

⇒ General base for *N*-Body simulations

https://github.com/AutoPas/AutoPas

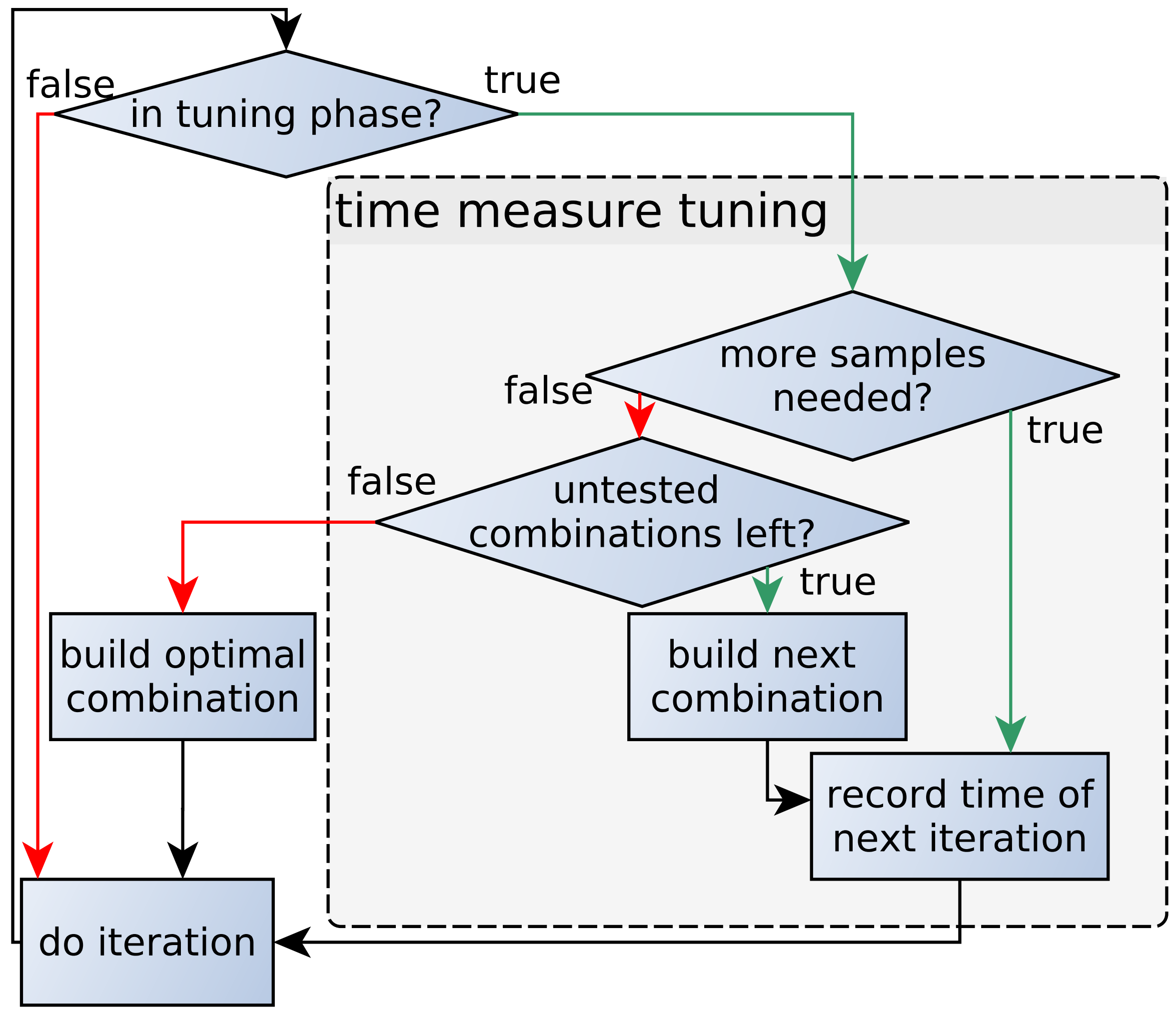


AutoPas: (Some) Implemented Options



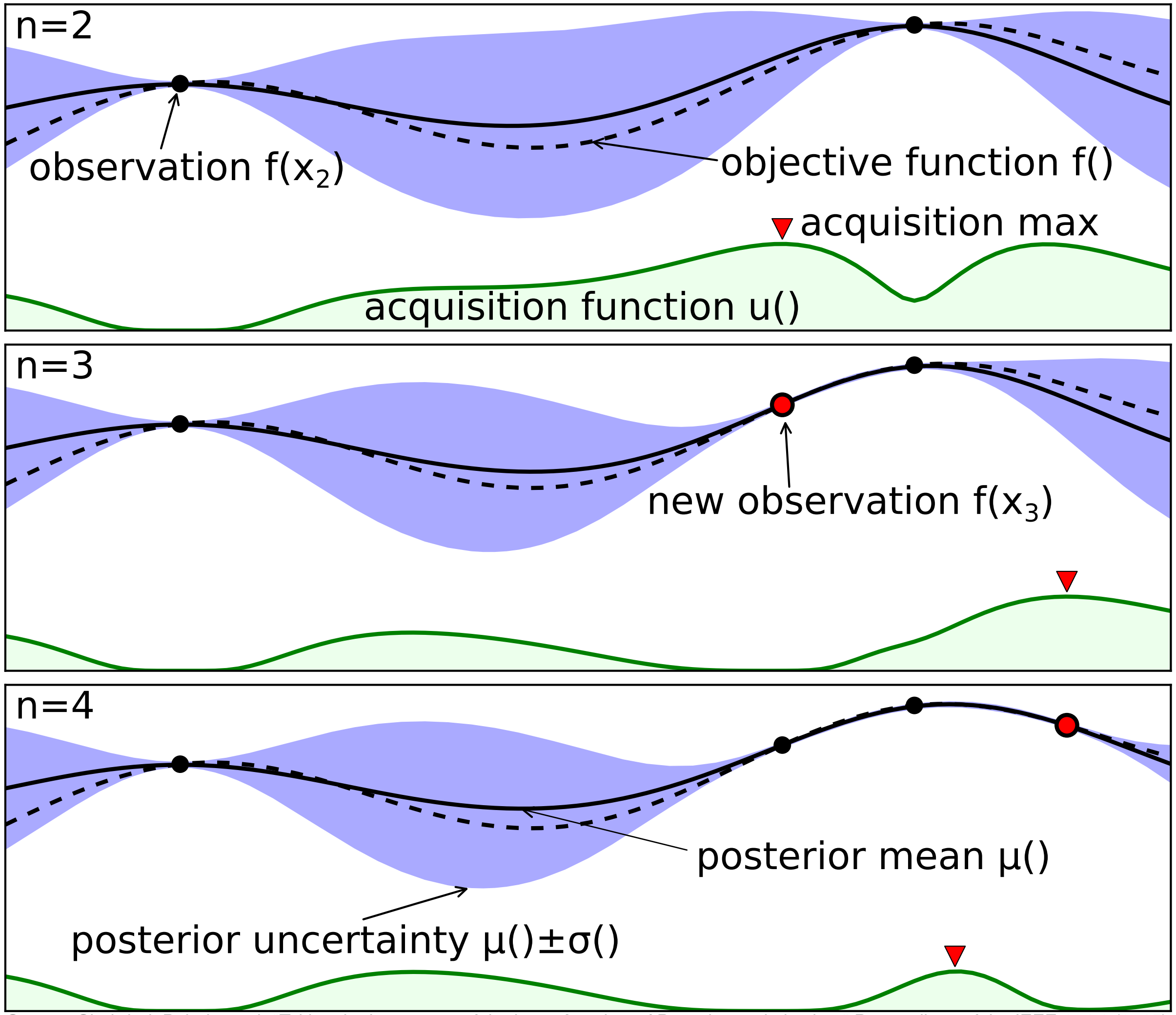
AutoPas: Auto-Tuning

- Common interfaces for containers, traversals, etc
  ⇒ Strategy pattern
  ⇒ "Verlet-like" approach
- Repeated periodically
- User can restrict search space



Auto-Tuning Strategies WiP - Bayesian Inference

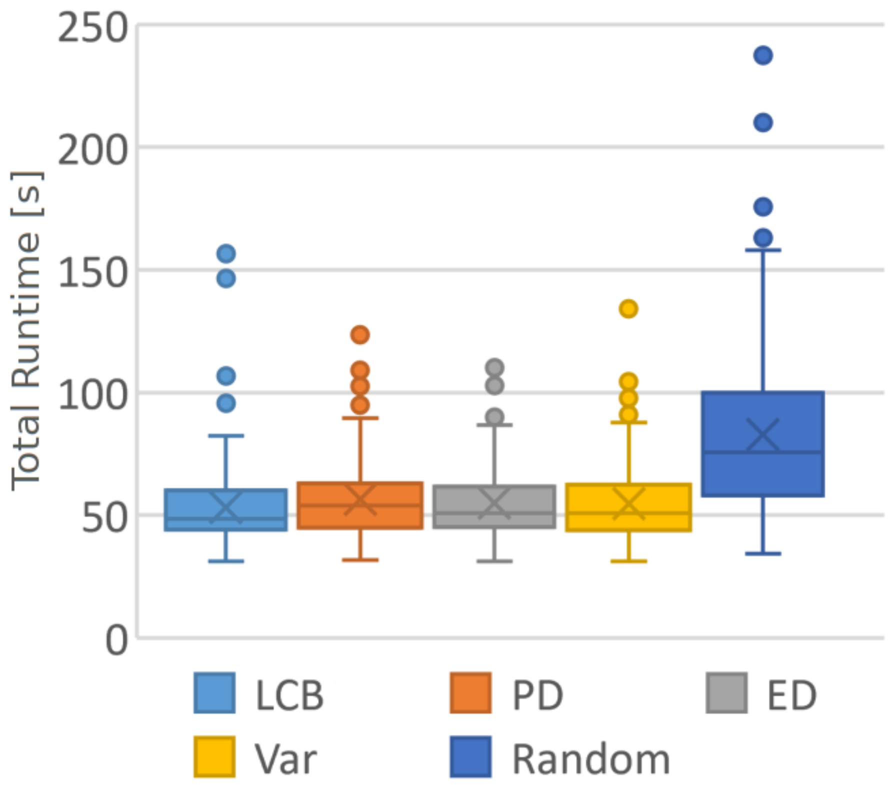
- Objective function:
  $\Rightarrow T_{iteration}(c)$
- Acquisition function:
  $\Rightarrow$?



Source: Shahriari, Bobak, et al. "Taking the human out of the loop: A review of Bayesian optimization." Proceedings of the IEEE 104.1 (2015)

Auto-Tuning Strategies WiP - Bayesian Inference

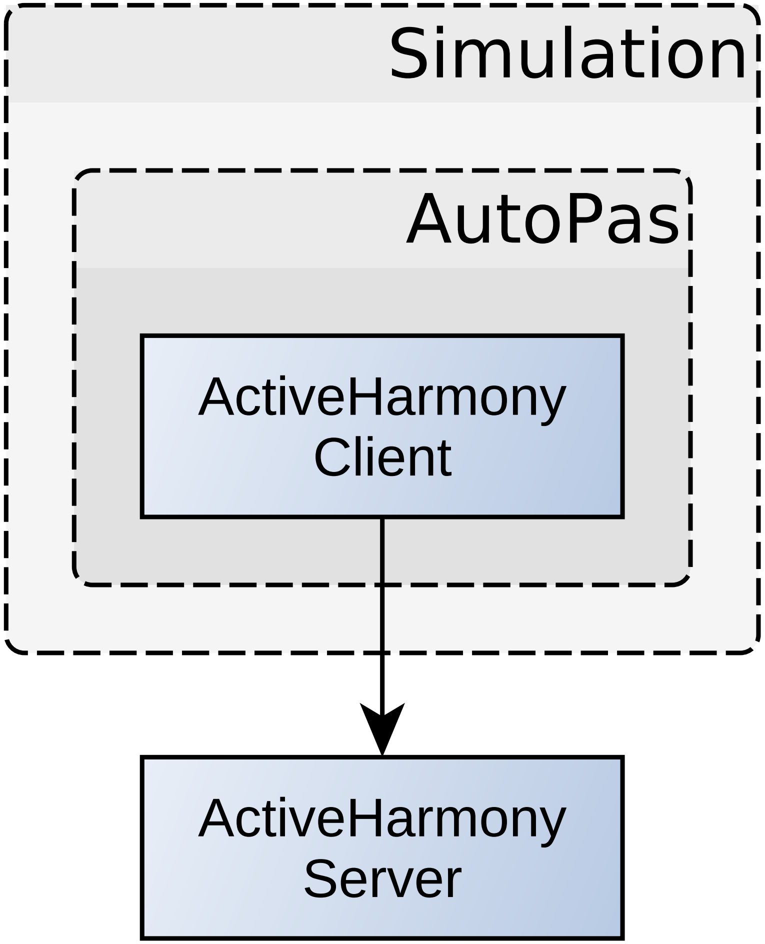
- 10 evidences sufficient instead of > 70
  $\Rightarrow$ Less iterations spent in tuning.
  $\Rightarrow$ Less testing of inefficient configurations.
- Here: No major differences in acquisition functions.



Scenario: 150k Particles densely Gaussian distributed; 50 iterations; 200 runs per column

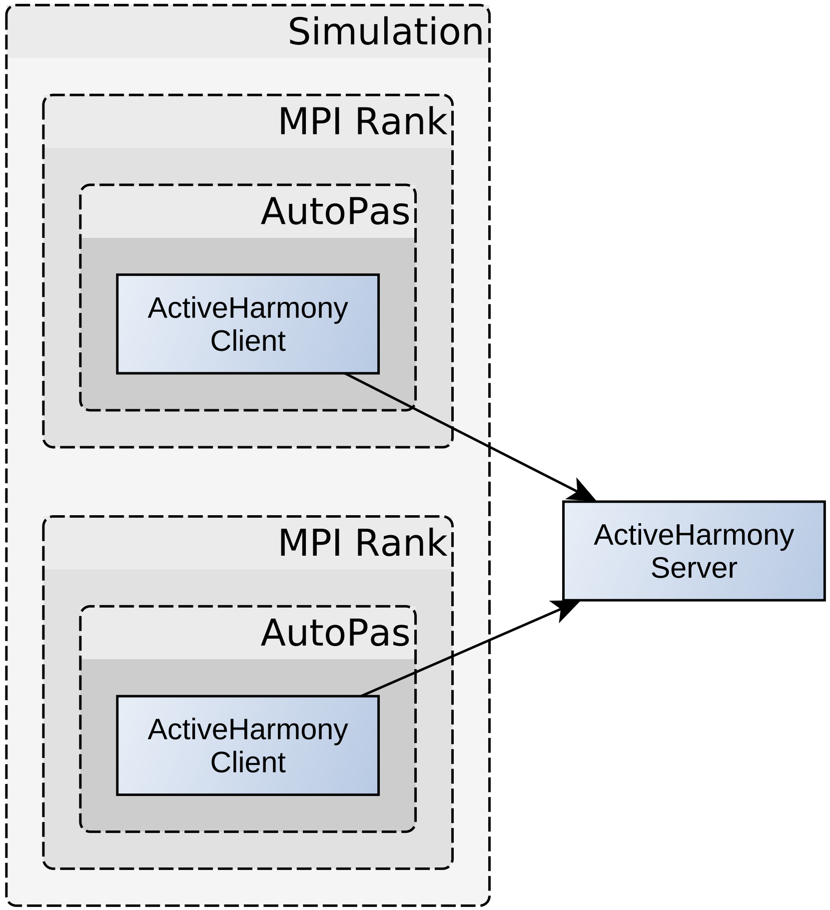
Auto-Tuning Strategies WiP - Active Harmony

- Open Source Auto-Tuning Framework
  https://www.dyninst.org/harmony
- Client-Server system
- Internally: Nelder Mead Simplex



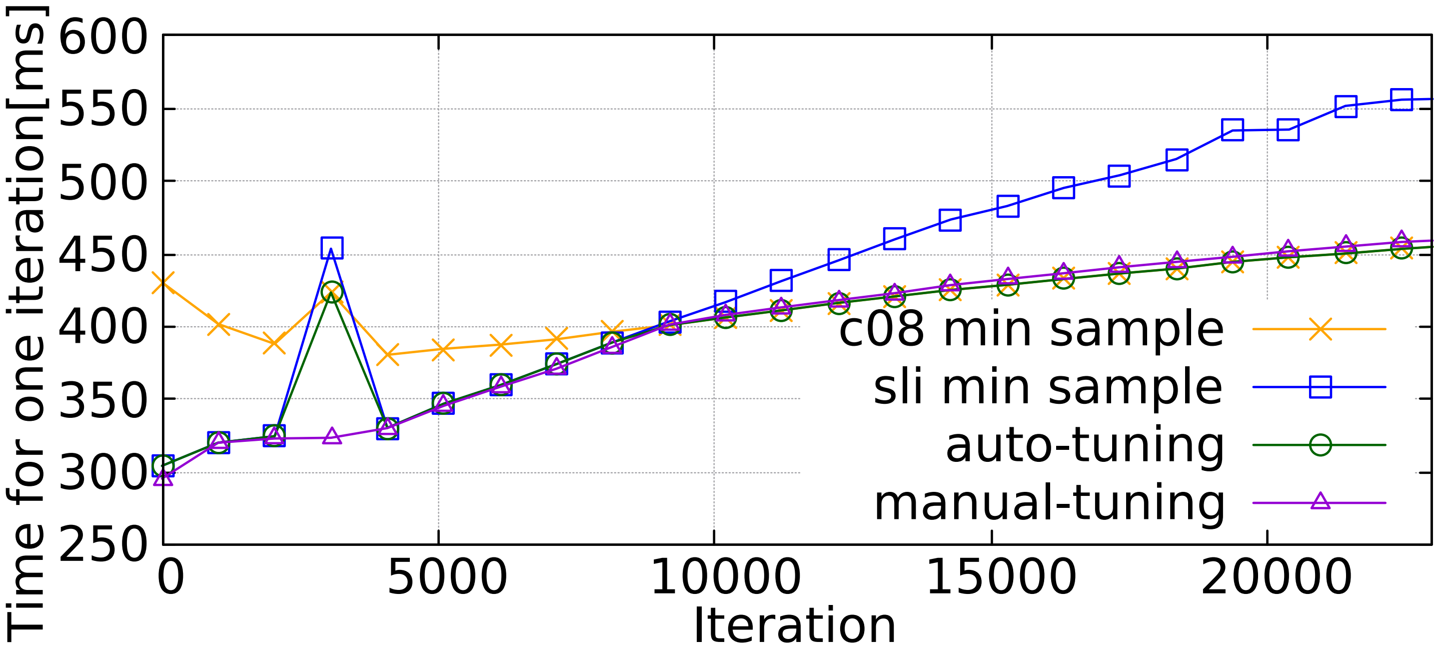
Auto-Tuning Strategies in Future - Distributed Search

- Spread Tuning over MPI Ranks
- Collect all tuning information in one server
- Greatly reduces number of tuning steps



Auto-Tuning Strategies in Future - Performance Extrapolation

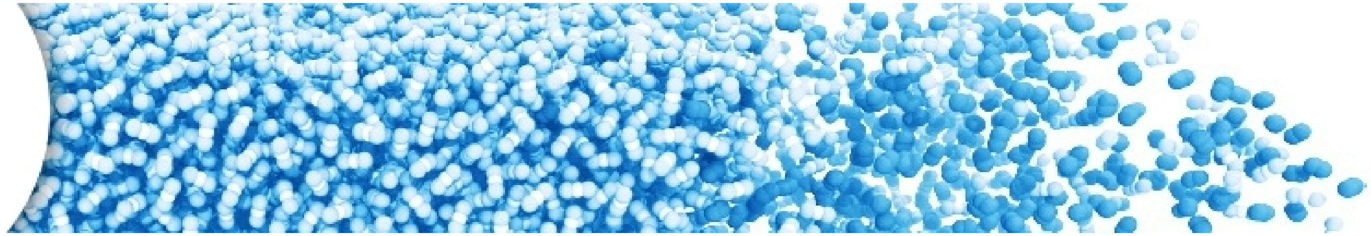
- Start with full search
- Extrapolate performance from previous evidences
- Decide if testing is worthwhile
- Easier Outlier decection



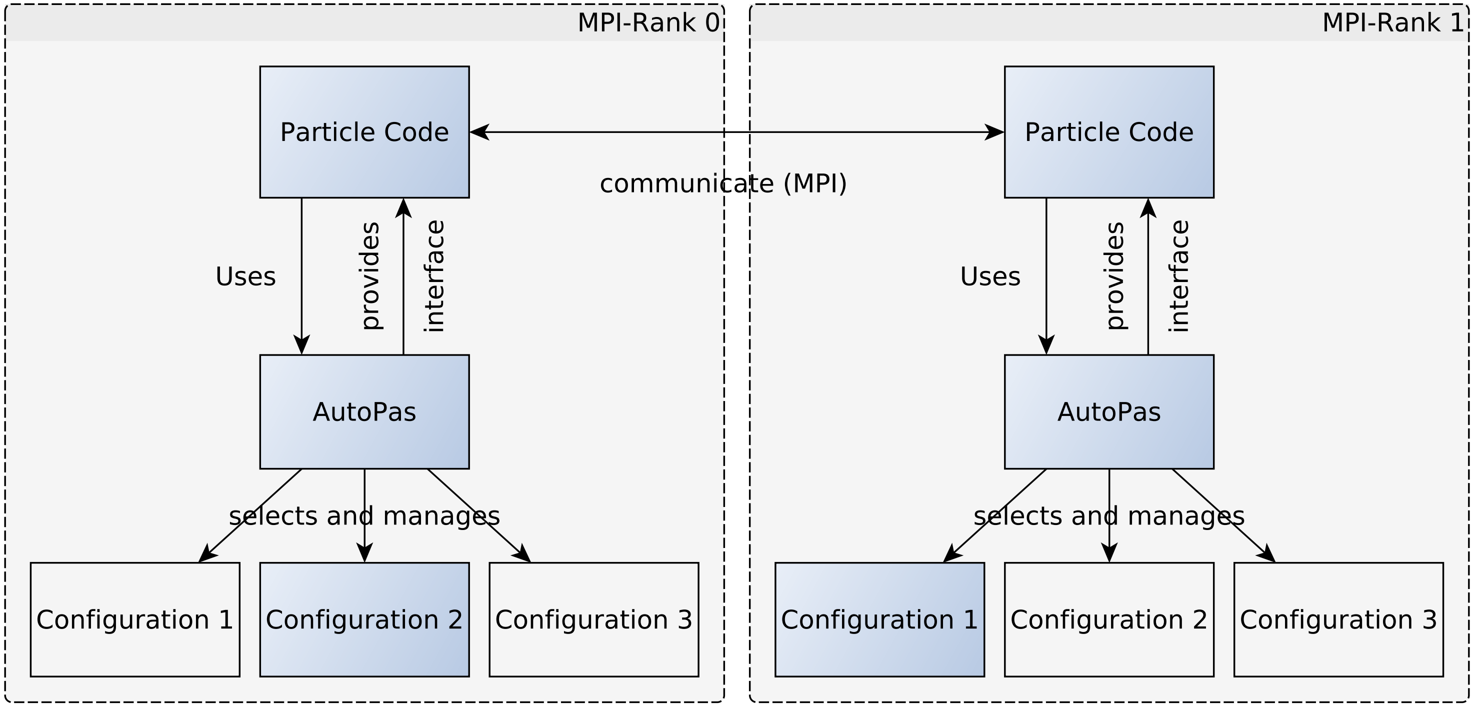
Actual Applications and Results

Integrating AutoPas into ls1 mardyn

- ls1 mardyn:
  - Large number of small rigid molecules.
  - Linked Cells Based.
  - MPI + OpenMP hybrid parallelization.
  - 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.

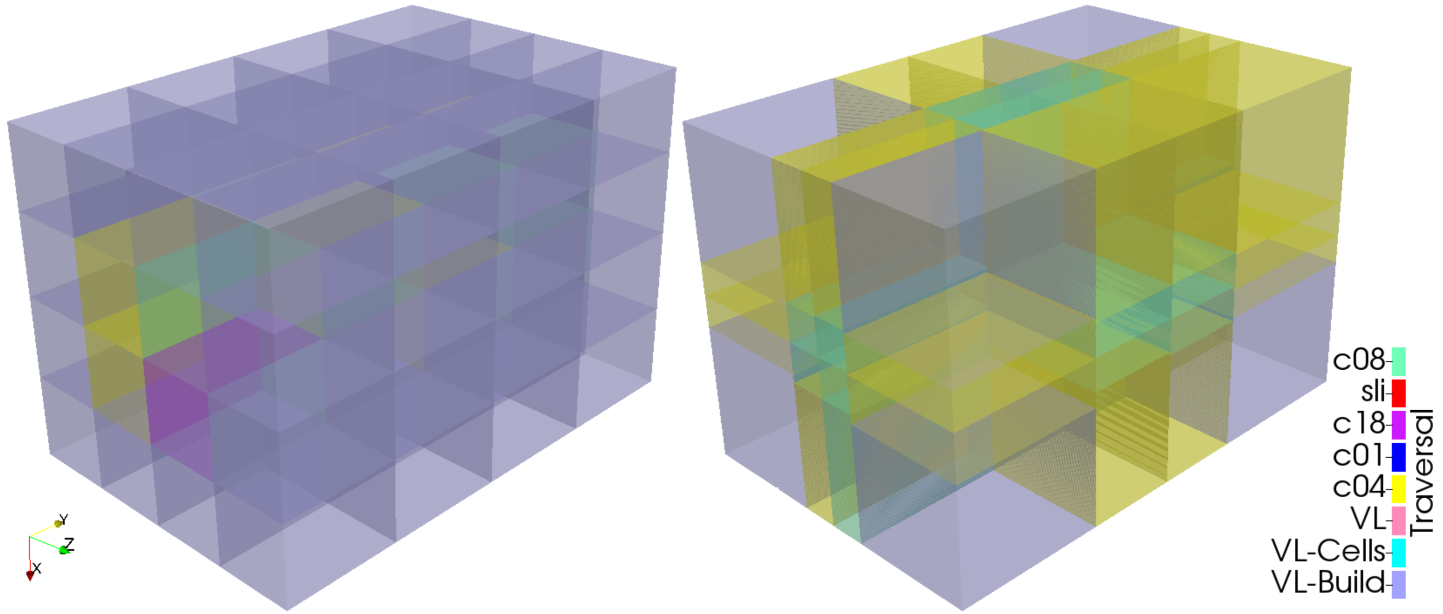


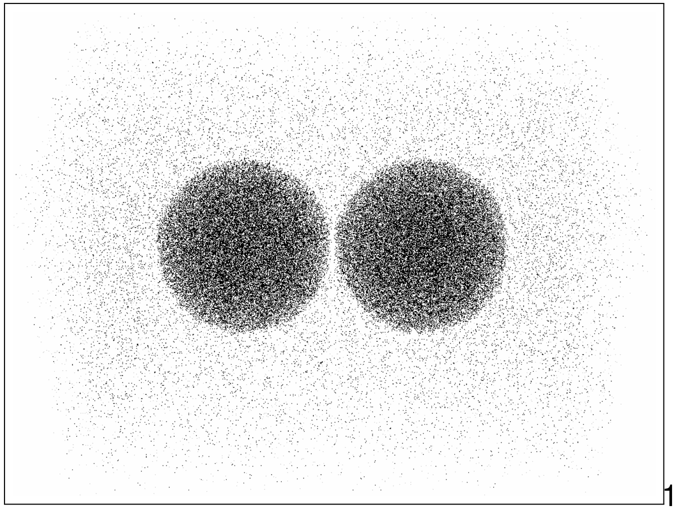
Integrating AutoPas into ls1 mardyn



- One AutoPas instance per Rank.
- Independent tuning!
- Interfaces independent of algorithms.
- AutoPas acts as black-box container.

Experiment: Droplet Coalescence





- 64 Ranks; 3M particles
- Auto-Tuning + Diffusive Loadbalancing
- Speed-up about 50% vs vanilla ls1

Conclusions

- AutoPas is a black box *N*-Body container.
- Dynamic tuning enables optimal performance for changing scenarios.
- Independent tuning over MPI parallel simulation.
- Achievable for users without expert knowledge.

... and future work

- New Auto-Tuning strategies.
- Tuning for more parameters and algorithms.

https://github.com/AutoPas