
Seminar Final Presentation

Algorithm Selection and Auto-Tuning in AutoPas

Manuel Lerchner
manuel.lerchner@tum.de

Table of Contents

- 1 Introduction
- 2 Background MD Algorithms
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

Software Challenges in Molecular Dynamics

- Enormous numbers of particles
 - naively: $\mathcal{O}(N^2)$ interactions
- Solution: highly optimized algorithms
 - Make full use of hardware
 - Smart data structures
- Every millisecond counts

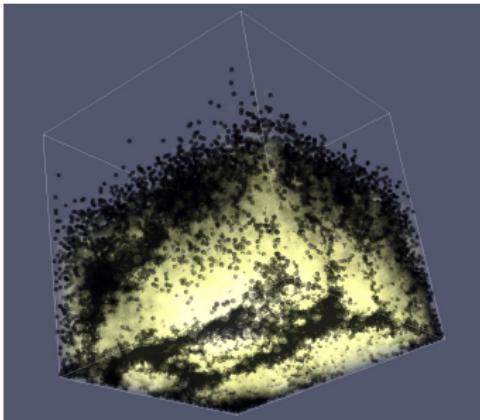
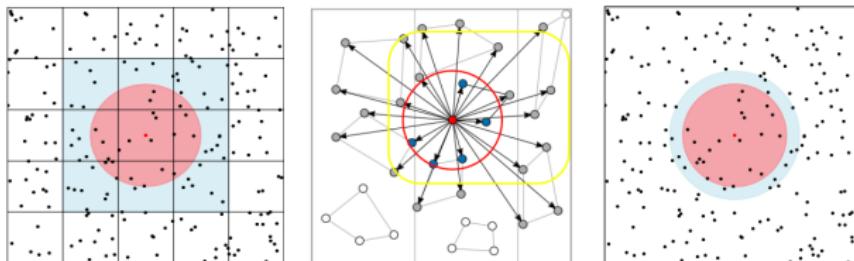


Table of Contents

- 1 Introduction
- 2 Background MD Algorithms
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

Example: Particle Container

- How to identify interacting particles?
- Short-range forces allow for cutoff radius $\rightarrow \mathcal{O}(N)$ interactions
- Different implementations possible



Linked Cells

Verlet Cluster Lists

Verlet Lists

Simpler Memory Access

Lower Memory Overhead

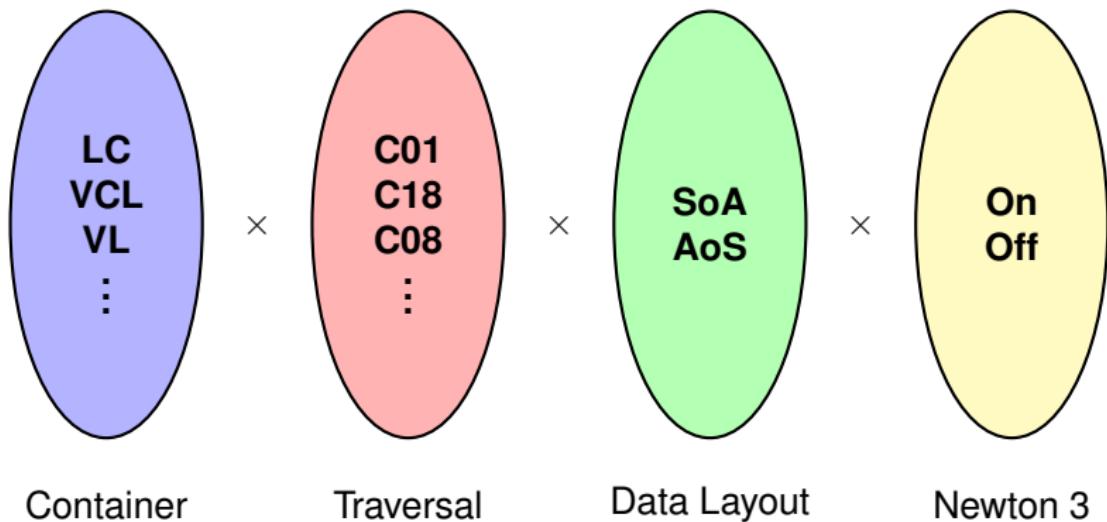
Fewer redundant calculations

Many difficult choices

- Shared memory traversals
 - Idle Time vs. Thread Overhead vs. Synchronization
- Data layouts
 - Good Vectorization (SoA) vs. Random Access (AoS)
- Newton 3 optimization
 - Reduce work vs. Bookkeeping overhead
- Depends on
 - Hardware
 - Scenario

Which Combination to Choose?

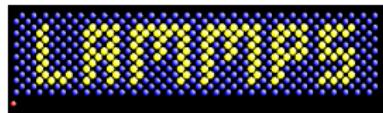
- Remember: No silver bullet!



Traditional MD Engines



- Based on a single combination
- Performance through:
 - Hardware aware SIMD vectorization
 - Memory layout optimization
 - Manual tuning to target hardware
- Drawbacks:
 - Suboptimal for some scenarios
 - Manual tuning effort



ls1
Mardyn

Table of Contents

- 1 Introduction
- 2 Background MD Algorithms
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

AutoPas

What is AutoPas?

- Library for arbitrary N-body simulations
- Implements all algorithms
 - Modular
 - All* configurations possible
- Idea: AutoTuning
 - Switch algorithms at runtime
 - Different strategies available
 - Thus optimal performance

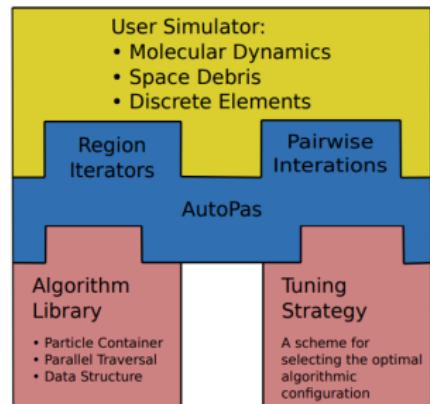


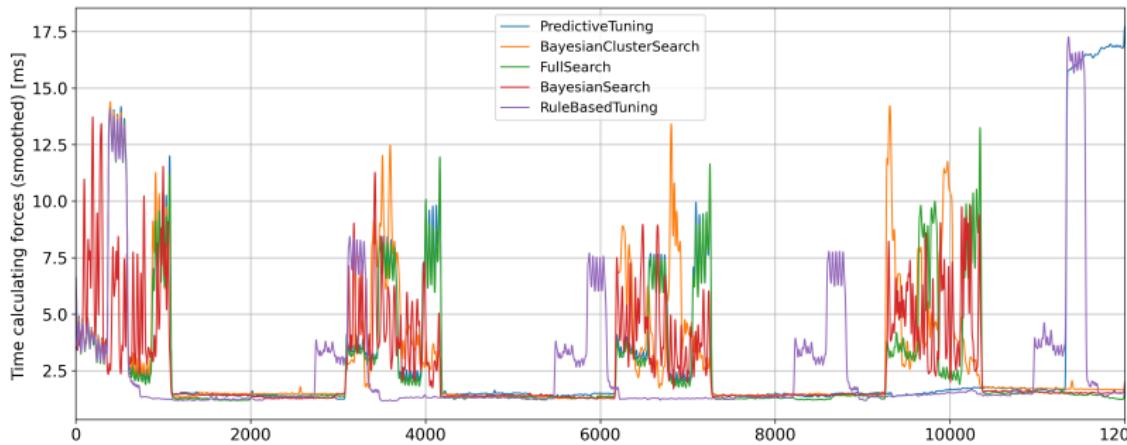
Figure: [Newcome et al., 2023]

Table of Contents

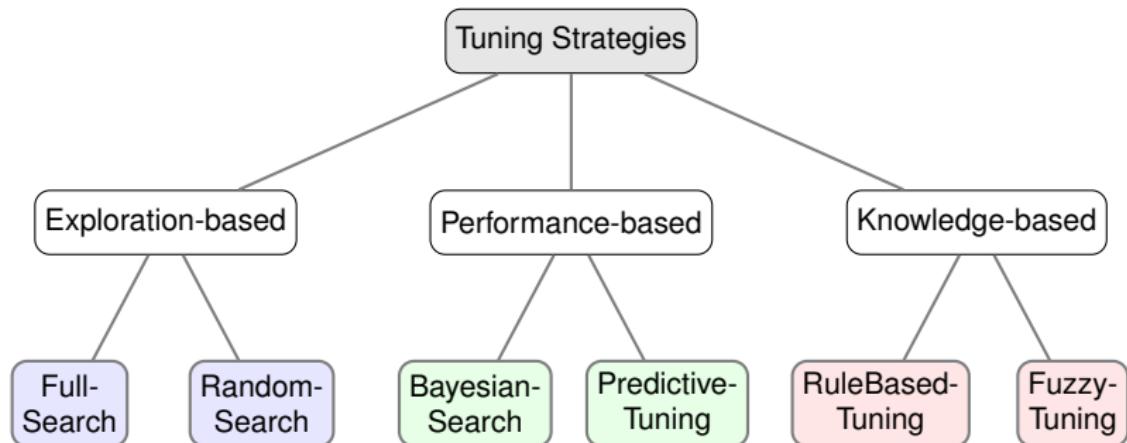
- 1 Introduction
- 2 Background MD Algorithms
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

Auto-Tuning in AutoPas

- Divide simulation into Tuning- and Simulation-Phase
- Tuning-Phase:
 - Tuning Strategies suggest configurations
 - Measure promising configurations
 - Choose best configuration



Tuning Strategies



Benefits of Auto-Tuning

- Optimal configuration
 - for every scenario
 - for every hardware
 - throughout the simulation
- Shown to improve performance in various scenarios
- User friendly: No manual tuning effort

Problems of Auto-Tuning

- Overhead from evaluating suboptimal configurations
 - Orders of magnitude slower!
- Unnecessary periodic re-tuning

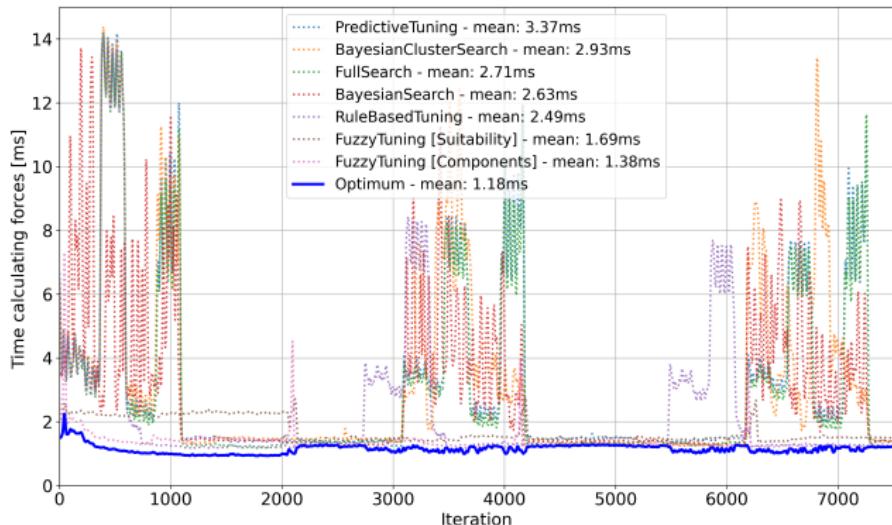
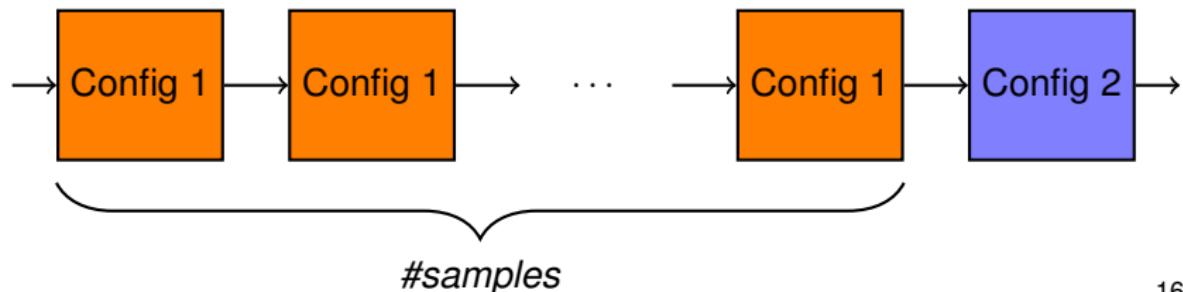


Table of Contents

- 1 Introduction
- 2 Background MD Algorithms
- 3 AutoPas Framework
- 4 Auto-Tuning in AutoPas
- 5 Early Stopping Optimization

Early Stopping Idea

- Optimization: Stop long-running evaluations early
- Configurations get sampled multiple times
- Two possible strategies:
 - Stop after first poor sample
 - Stop during evaluation

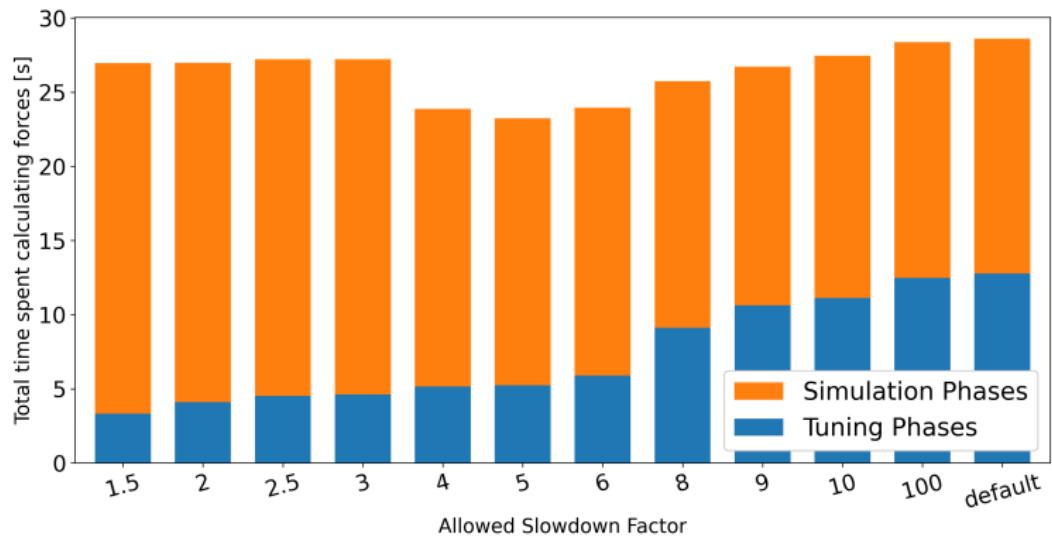


Naive Early Stopping

- Changes to AutoPas are minimal
 - Keep track of best performance seen so far
 - $slowdownFactor = \frac{t_{\text{sample}}}{t_{\text{best}}}$
 - Call `retune()` if $slowdownFactor$ exceeds threshold
- Hyperparameter: $allowedSlowdownFactor$
- Which $allowedSlowdownFactor$ should be used?
 - $allowedSlowdownFactor \rightarrow 1$: Many spurious aborts
 - $allowedSlowdownFactor \rightarrow \infty$: No early stopping

Benchmark: Exploding Liquid + Predictive Tuning

- Optimal at $allowedSlowdownFactor = 5$
- From 28.6s to 23.2s
 - 18.9% reduction



Early Stopping Discussion

- Should be combined with good tuning strategies
- Never increased simulation time
- *allowedSlowdownFactor* probably not universal
- Future work needed
 - More benchmarks
 - Investigate other stopping criteria

Thank you for your attention!

Questions?

References I

-  Newcome, S. J., Gratl, F. A., Muehlhaeusser, M., Neumann, P., and Bungartz, H.-J. (2024). Autopas: Dynamic algorithm selection in molecular dynamics for optimal time and energy.
In SIAM Conference on Parallel Processing (PP24). SIAM.
-  Newcome, S. J., Gratl, F. A., Neumann, P., and Bungartz, H.-J. (2023). Towards the smarter tuning of molecular dynamics simulations.
Amsterdam, The Netherlands.