Chair of Scientific Computing
Department of Informatics
Technical University of Munich

150 Jahre culture of excellence

AutoPas: A Library for N-Body Simulations Enabling Optimal Node-Level Performance Through Auto-Tuning

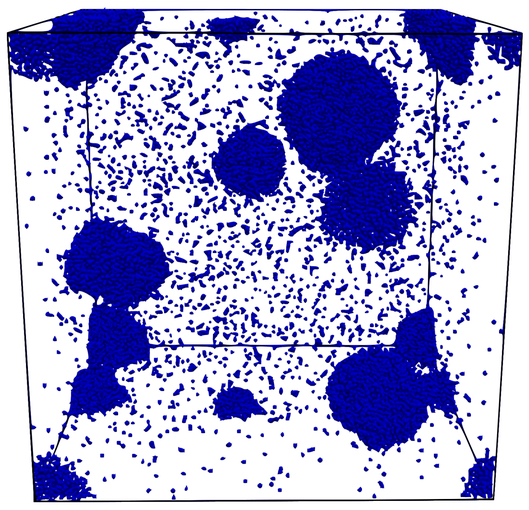
Fabio Gratl[†] Steffen Seckler[†] Nikola Tchipev[†] Philipp Neumann[‡] Hans-Joachim Bungartz[†]
[†]{f.gratl,s.seckler,n.tchipev,bungartz}@tum.de, Chair of Scientific Computing in Computer Science, Technical University of Munich
[‡]philipp.neumann@uni-hamburg.de, Group Scientific Computing, Universität Hamburg

Molecular Dynamics

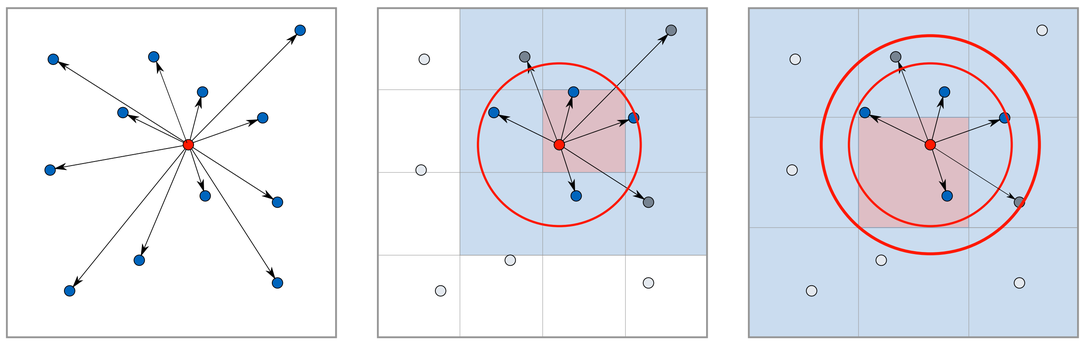
Applications:
- Chemical Engineering: cavitation, surface tension, droplet coalescence, etc...

Goals:
- High node-level performance in arbitrary scenarios.
- Minimize time to solution.

Main Challenges:
- Drastic impact of simulation variables on time to solution.
- Performance depends on many variables that can change during runtime.

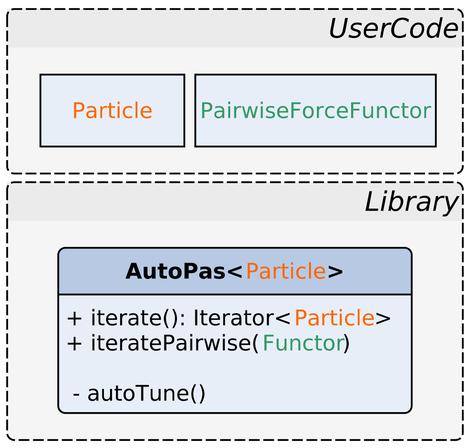


Motivation



- Several data structures, algorithms, parallelizations, etc. available.
- All have scenario dependent strengths.
- Choosing optimal combination not always straight forward.

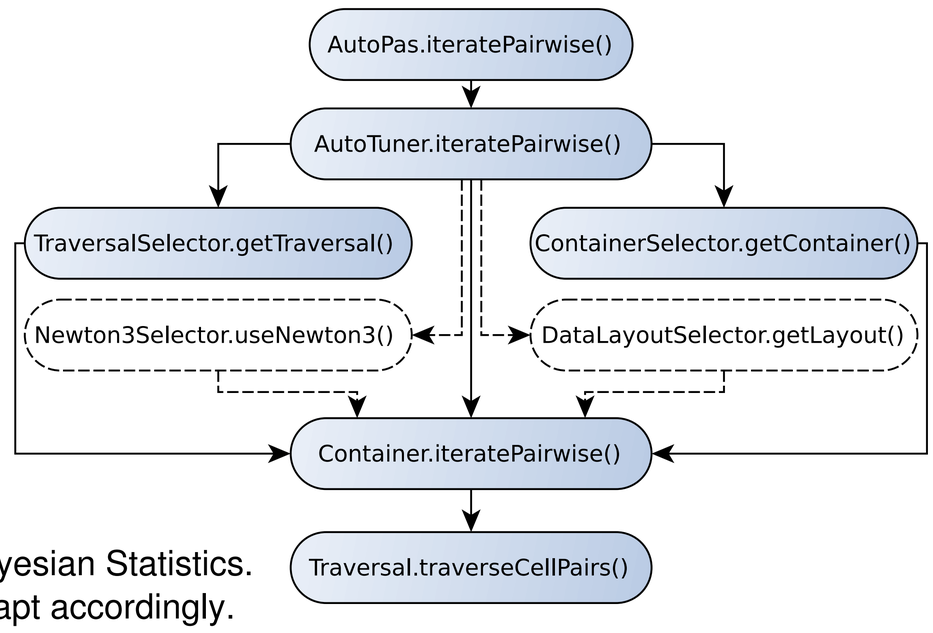
AutoPas Library



AutoPas' Vision:
- Base to build full N-Body simulations on top of.
- Manages node-level performance internally via Auto-Tuning.
- Modular C++ template design able to dynamically select optimal SIMD, OpenMP, data structures, etc. at runtime.

Auto-Tuning:
- Instead of the user, the code should find the optimal algorithms.
- Too many combinations possible to test all.

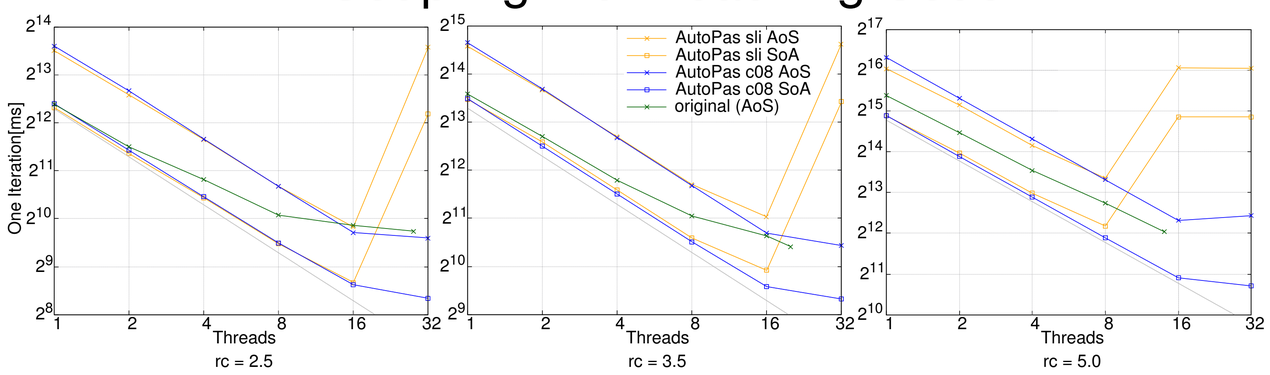
⇒ Outlook:
Performance Modeling, Machine Learning, Bayesian Statistics.
- Reevaluate combination during runtime and adapt accordingly.
- Provide flexibility through "strategy" software pattern.



User Defines:
- Particle class describing physical properties of single particles.
- Pairwise force functors describing single interactions of particles.

First Results

Coupling with Teaching Code



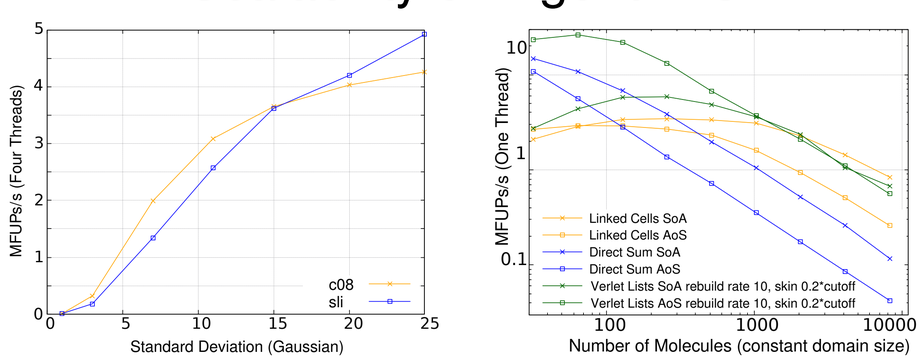
An early version of AutoPas was coupled with a teaching code by students. This was done to test the performance as well as verifying usability.

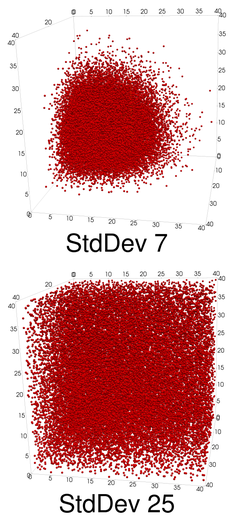
Scenario:
- Body-centered cubic lattice with 2M particles.
- Strongscaling on SuperMUC Phase 1.
- 54 Linked Cells per dimension (excl. halo) ≈13 particles per cell.

Observations:
- Performance of algorithms varies significantly.
- For every configuration at least one algorithm outperforms the original code.

⇒ Target for Auto-Tuning!
- AutoPas can offer near optimal scaling.

Scalability of Algorithms





Scenario:
- Homogenity scaling.
- 8M particles.
- Gaussian distribution.
- 100 cells per dim.
- 8 Threads

Observations:
- c08 (8-way coloring)

⇒ Good load balancing.
- sli (1D partitioning)

⇒ Minimal overhead.

Scenario:
- Density scaling.
- Container comparison.
- Uniform distribution.
- 10 cells per dim.
- No parallelization.

Observations:
- See Motivation.
- Linked Cells are more efficient in high densities.

References
[1] N. Tchipev, A. Wafai, C. W. Glass, W. Eckhardt, A. Heinecke, H.-J. Bungartz, and P. Neumann, "Optimized force calculation in molecular dynamics simulations for the intel xeon phi," in European Conference on Parallel Processing, pp. 774–785, Springer, 2015.
[2] N. Tchipev and et al., "Twetris: Twenty trillion-atom simulation." submitted, 2018.
[3] N. Tchipev, A. Costinescu, S. Seckler, P. Neumann, and H.-J. Bungartz, "Towards autotuning between openmp schemesfor molecular dynamics on intel xeon phi," 2017. SIAM CSE '17.

Bundesministerium für Bildung und Forschung
Intel Parallel Computing Center TUM & LRZ
We thank the Intel Parallel Computing Center "ExScaMIC-KNL" and the Federal Ministry of Education and Research, Germany, project "Task-based load balancing and auto-tuning in particle simulations" (TaLPas), grant number 01IH16008 for financial support of this research.

github.com/AutoPas
gratl @ sccs