Chair of Scientific Computing in Computer Science
School of Computation, Information and Technology
Technical University of Munich

Towards the Smarter Tuning of Molecular Dynamics Simulations

Samuel J. Newcome[1]*, Fabio A. Gratl[1], Philipp Neumann[2], Hans-Joachim Bungartz[1]

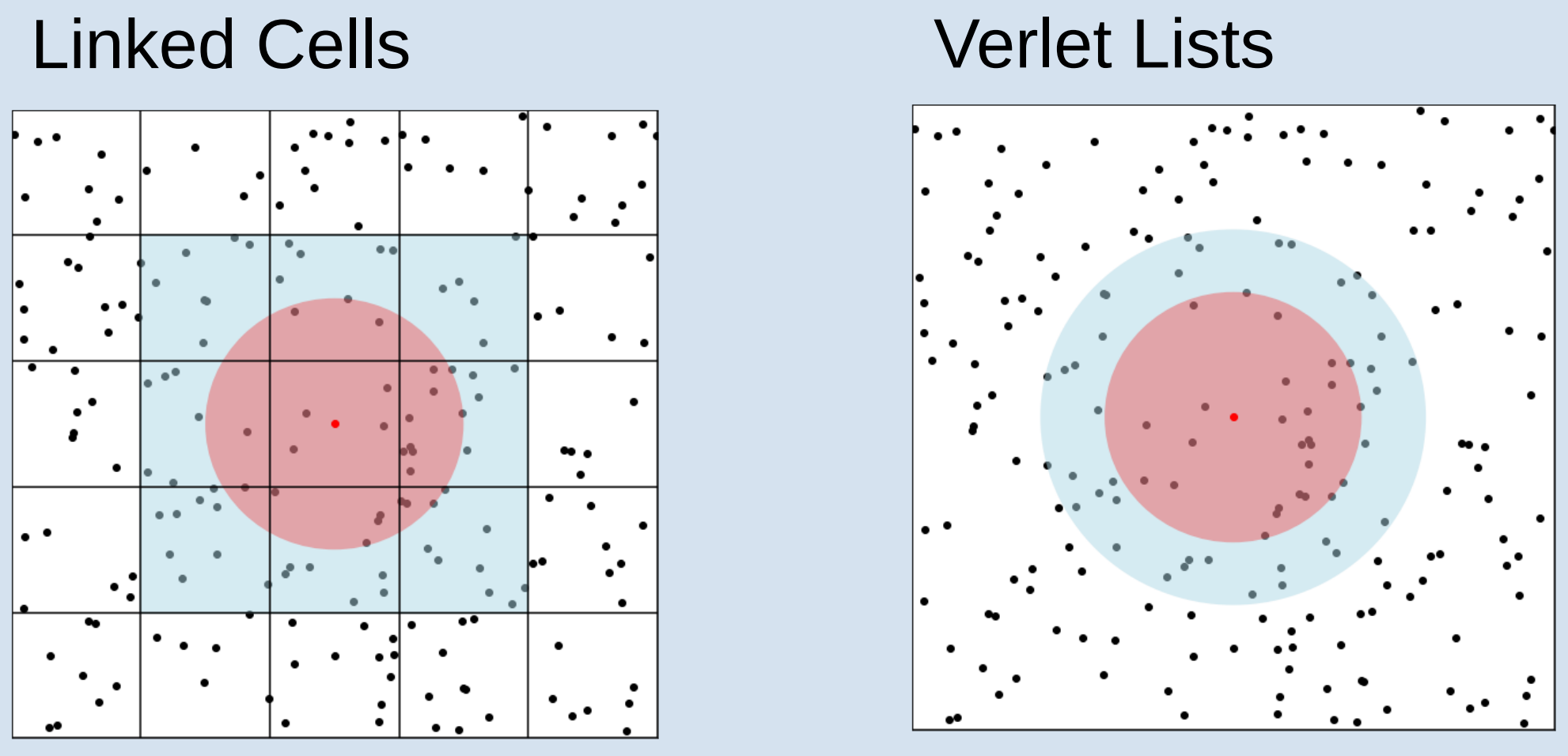
[1] Chair for Scientific Computing in Computer Science; School of Computation, Information and Technology; Technical University of Munich
[2] Chair for High Performance Computing; Fakultät für Maschinenbau; Helmut-Schmidt-Universität
* samuel.newcome@tum.de

AutoPas: A Node-Level Auto-Tuning Library for Short-Range Particle Simulations

Many algorithms exist for the efficient evaluation of short-range pairwise interactions like the Lennard-Jones force in large particle simulations. AutoPas contains implementations of many of these. Each has its strengths and weaknesses, and with different optimal algorithms in different scenarios.

Choosing the best algorithm is a very hard task, and the best algorithm can change throughout the simulation. AutoPas aims to select the best algorithm dynamically throughout the simulation.

Particle Containers

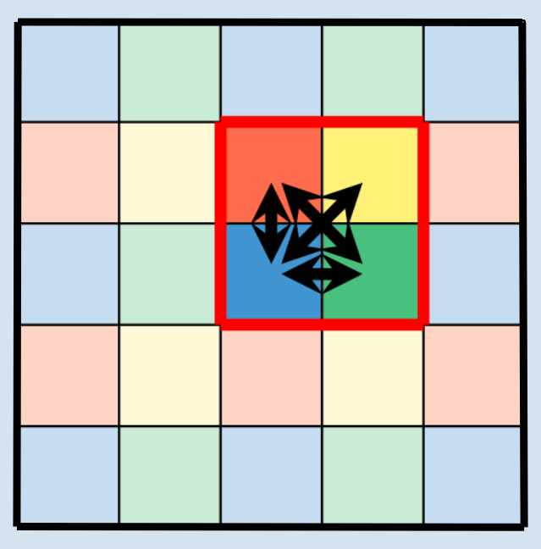


Linked Cells:
+ Low overhead
+ Easily vectorised
- Low Hitrate

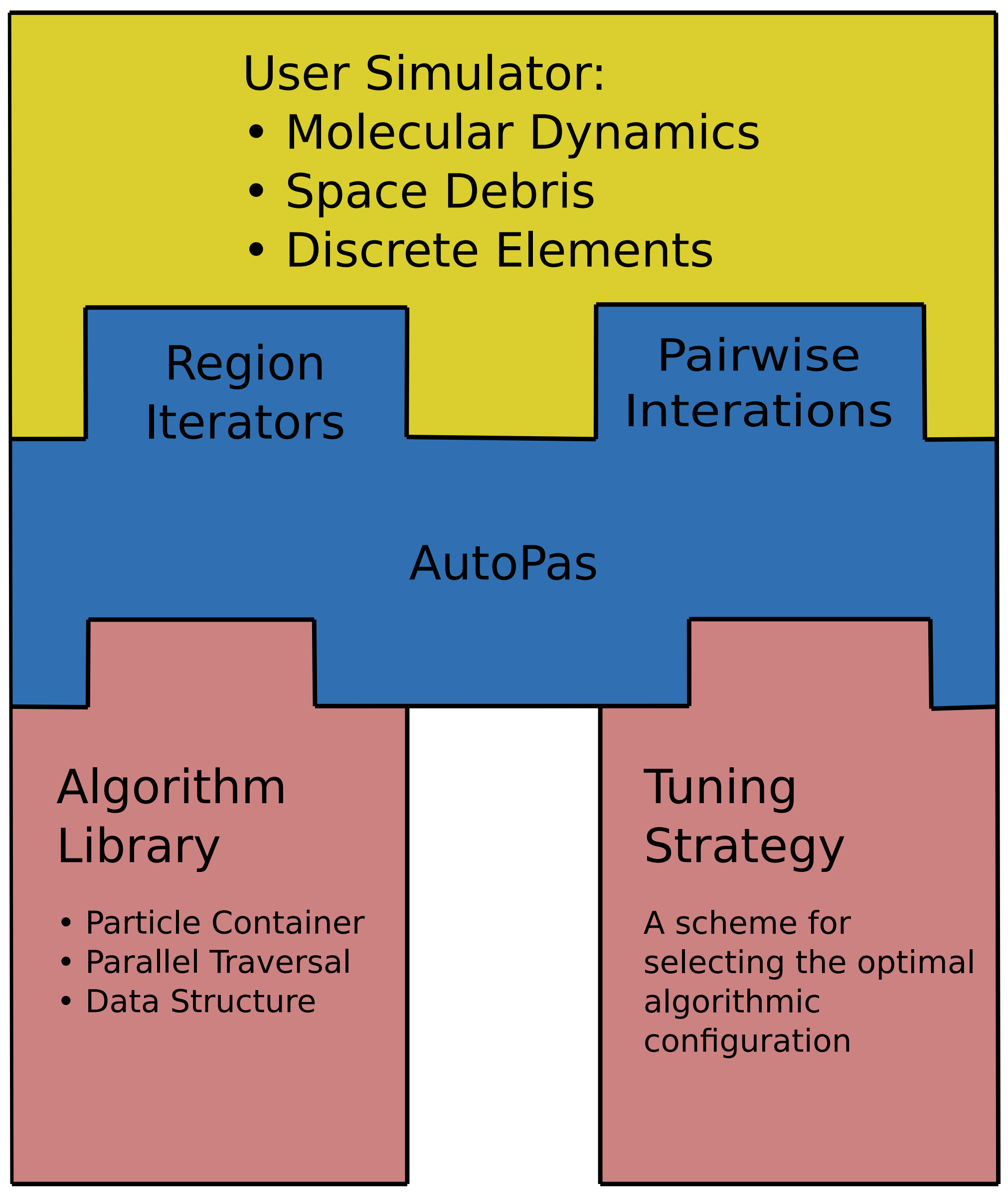
Verlet Lists:
+ High Hitrate
- High overhead
- Poor vectorisation

Parallel Traversals

The choice of how to traverse every pair of particles in parallel to use effects:
- load balance
- scheduling
- thread idle times



Example: C08 Scheme



User Simulator:
- Molecular Dynamics
- Space Debris
- Discrete Elements

Region Iterators

Pairwise Interations

AutoPas

Algorithm Library
- Particle Container
- Parallel Traversal
- Data Structure

Tuning Strategy

A scheme for selecting the optimal algorithmic configuration

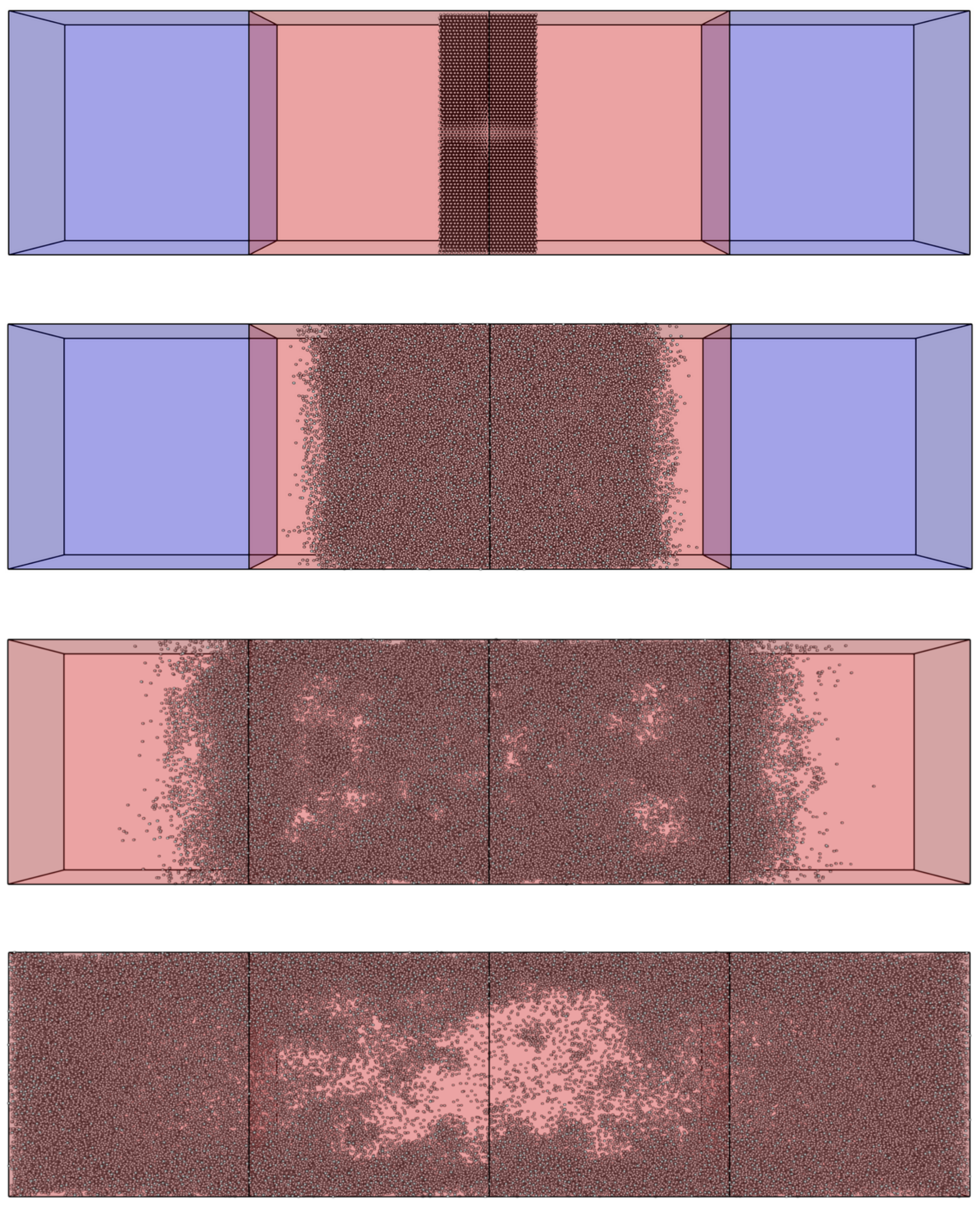
Tuning Strategies

Testing poorly performing algorithmic configurations wastes time ⇒ A good tuning strategy balances the potential to choose a better algorithmic configuration with the cost of testing algorithms worse than the current configuration.

- Single Full Search: Run a full search over the entire algorithm library and choose the best.
- Multiple Full Search: Run multiple full searches at intervals.
- Student Developed Predictive Tuning:
  - Fit a model for every configuration based on past data.
  - Simple models, like fitting a line through the last two data points, work best.
  - Trial, at intervals, algorithms with predicted performances near the optimal prediction.
  - Trial, at much longer intervals, algorithms that have not been trialled recently.
  - Blacklist terrible performing algorithms.

- Future Strategies:
  - Bayesian Optimisation Based Approaches.
  - Using MPI to learn optimal algorithms from domains.

Application: An Exploding Liquid



Optimal Algorithms

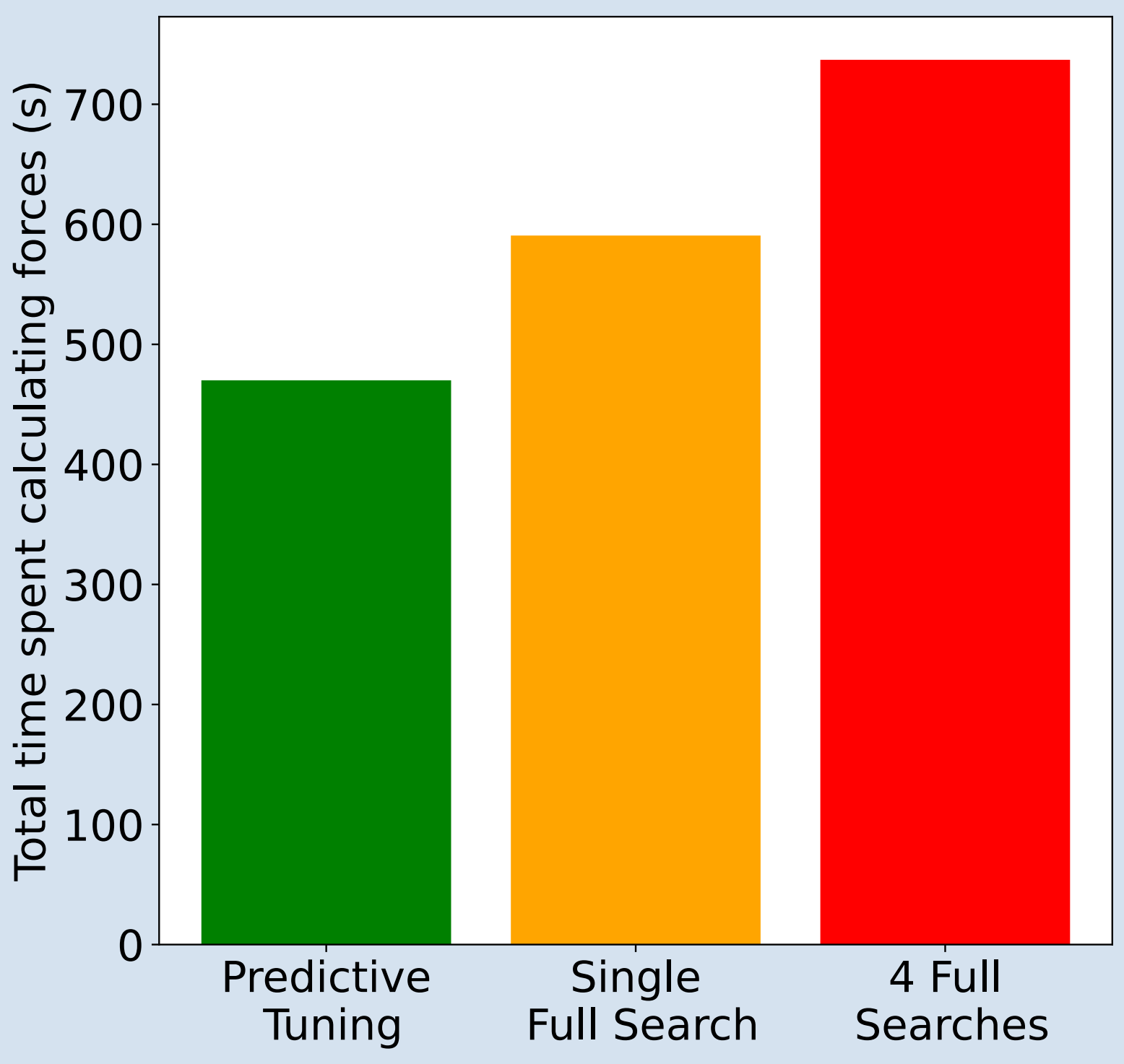
🟥 Container: Linked Cells
Traversal: Sliced C02
Data Structure: Array-of-Structures

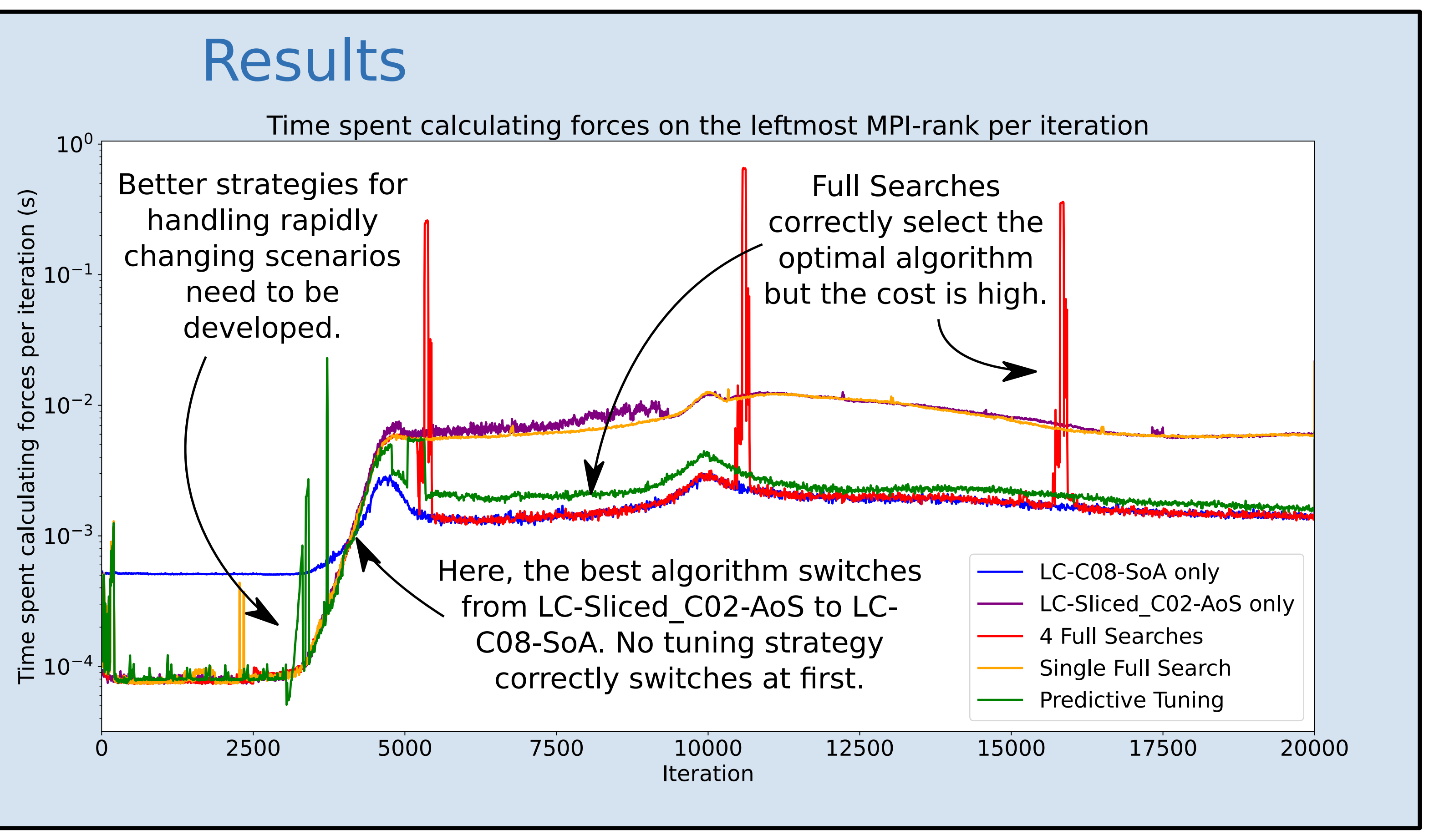
🟦 Container: Linked Cells
Traversal: C08
Data Structure: Structure-of-Arrays

Scenario:
- 60x240x60 domain, 4 MPI x 36 OMP.
- Run on 2 compute nodes of HSUper, each with 256GB RAM and 2 Intel Icelake Socket, with each Socket having an Intel(R) Xeon (R) Platinum 8360Y processor with 36 cores.
- Periodic Boundary Conditions.
- 124320 tightly packed Lennard-Jones Molecules placed in the centre of the domain.
- The Lennard-Jones fluid 'explodes' out into the otherwise empty domain, resulting in a sparse, inhomogeneously populated domain.
- Run for 20000 time-steps of Δt=0.001824

Tuning Strategies Tested:
- A Single Full Search.
- 4 Full Searches at equal intervals.
- Predictive tuning with simple line extrapolation:
  - Tuning intervals of 250 iterations.
  - ⇒ A total of 73 tuning phases.
  - Trialling algorithms predicted to perform less than 30% slower than predicted optimum, unless they haven't been tried in 120 tuning phases.
  - Blacklisting algorithms trialling more than 2x slower than the optimum.

Results




Time spent calculating forces on the leftmost MPI-rank per iteration

Better strategies for handling rapidly changing scenarios need to be developed.

Full Searches correctly select the optimal algorithm but the cost is high.

Here, the best algorithm switches from LC-Sliced_C02-AoS to LC-C08-SoA. No tuning strategy correctly switches at first.

References:
[1] Gratl, F.A., Seckler, S., Bungartz, H.J. and Neumann, P., 2022. N ways to simulate short-range particle systems: Automated algorithm selection with the node-level library AutoPas. Computer Physics Communications, 273, p. 108262.
[2] Gratl, F.A., Seckler, S., Tchipev, N., Bungartz, H.J. and Neumann, P., 2019, May. Autopas: Auto-tuning for particle simulations. In 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (pp. 748-757). IEEE.
[3] Seckler, S., Gratl, F., Heinen, M., Vrabec, J., Bungartz, H.J. and Neumann, P., 2021. AutoPas in ls1 mardyn: Massively parallel particle simulations with node-level auto-tuning. Journal of Computational Science, 50, p.101296.

Acknowlegements:
Developments are supported by the project "MaST: Macro/Micro-Simulation of Phase Separation in the Transcritical Regime" of the Digitalization and Technology Research Center of the Bundeswehr (dtec.bw).
Computational resources (HPC-cluster HSUper) have been provided by the project hpc.bw, funded by dtec.bw — Digitalization and Technology Research Center of the Bundeswehr.