MPI-OpenMP Load Balanced Simulation of Inhomogeneous Particle Systems in ls1 Mardyn at Extreme Scale

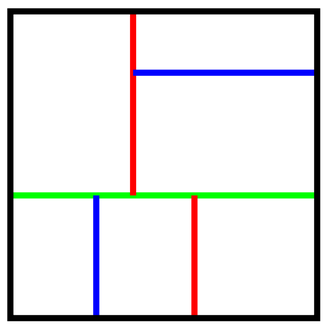
Steffen Seckler[1], Nikola Tchipev[1], Matthias Heinen[2], Fabio Gratl[1], Hans-Joachim Bungartz[1], Philipp Neumann[3]

[1]Technical University of Munich, [2]Technische Universität Berlin, [3]University of Hamburg

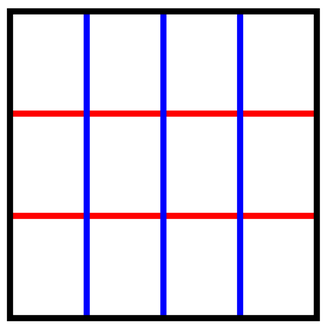
Introduction

- Highly-heterogeneous vapor liquid equilibrium simulation of the coalescence of two liquid droplets in a vapor phase
- MPI + OpenMP simulation
- Using ls1 mardyn – World Record Simulator (20 trillion particles) [2] – linked cell-based

Load Balancing and MPI



(a) Space-partitioning using kdd. The different colors represent the different levels of the splitting hyperplanes.



(b) Space-partitioning using a 2-d cartesian grid (sdd). Shown is a splitting into 12 subdomains.

Figure 1: Space partitioning algorithms supported by ls1 mardyn.

Scenario is highly heterogeneous so load balancing is necessary.

- Two partitioning algorithms:
  - Load balancing k-d tree-based decomposition (kdd)
  - Load unaware standard domain decomposition (sdd)
- Load estimation – all cell based

  old Simple cost model: $C_{\text{cell}} = n_{\text{cell}}^2 + \frac{1}{2}\sum_{\text{neighbors}} n_{\text{cell}} \cdot n_{\text{neighbor}}$

  vectuner Measure times on model problem, get C_{cell}.

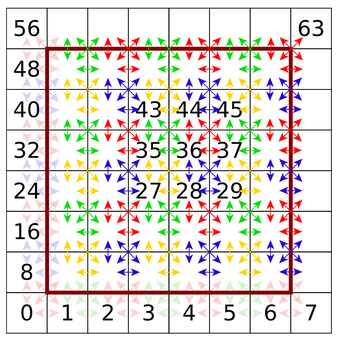
  measureLoadv1 Measure times T_i on real simulation: $T_i = \sum_{c \in \mathcal{C}} n_{\text{i,c}} \cdot C_{\text{cell}}$. Solve matrix equation.

  measureLoadv2 Same as measureLoadv1, fit quadratic function.

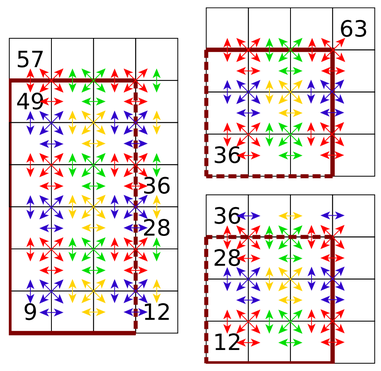
MPI Communication

- Uses eighth shell method [1].
- Uses overlapping global collectives.
- Communication partners are built via request principle and thus flexible in regards to zonal methods.

Eighth Shell Method



(a) Single node. Eighth Shell only opaque arrows, Full Shell also transparent.



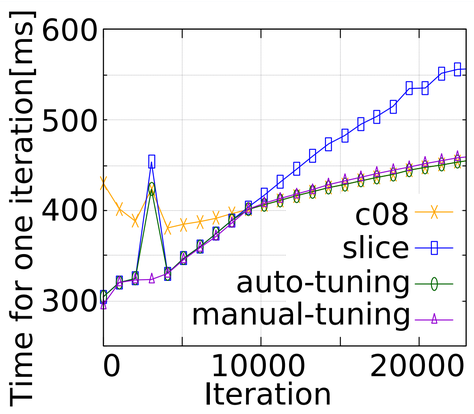
(b) Eighth shell method with a partitioning in 3 subdomains.

Figure 2: C08 traversal (left) and eighth shell partitioning (right).

- Enables newton-3 optimization across process boundaries.
- Adds force-exchange step in comparison to full-shell method.
- Based on c08-traversal.
- Proper partitioning of force calculations only if all cells are of the same size.

Outlook

- Integration of AutoPas
- Enables auto-tuning.
- MS319
  Molecular Dynamics
  4:10 PM - 5:50 PM
  Room: 303B
- MS319 includes talk on AutoPas (5pm - 5:20pm).



Scenario Results

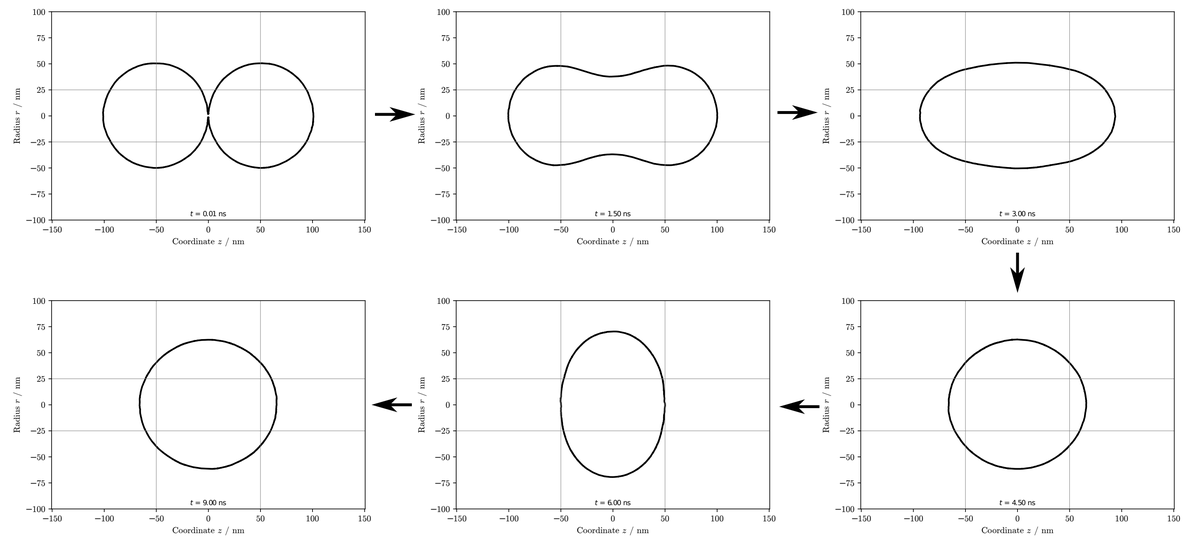


Figure 3: Shown is the contour of the liquid droplets for the d02 scenario over time.

- Three scenarios, total molecule count: 3M, 25M, 200M.
- Simulation time: 1 day on 128 nodes (3M), 5 days on 512 nodes (25M) and 15+ days on 1024 (200M)
- The coalescence of the droplets takes roughly twice the time if the droplet radius is doubled: Time until second state is reached: 0.7ns (3M) 1.5ns (25M), 3.8ns (200M)

Results

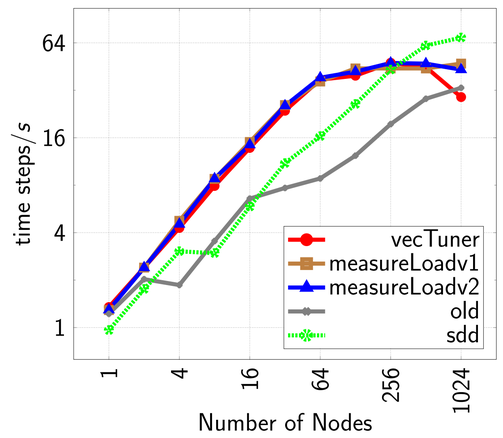


Figure 4: The different load estimation techniques for the d01 scenario with 8 OpenMP threads per rank using full shell.

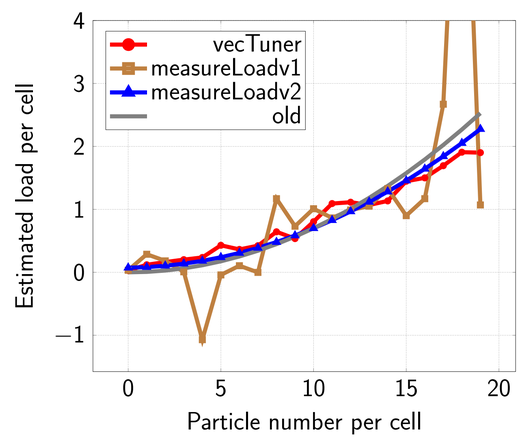


Figure 5: Estimated load of different load estimation techniques for the 3M scenario using full shell.

- Old: performs worst, underestimates load of sparsely populated areas.
- measureLoadv1: Good scaling, but clear outliers, where few data points exist.
- measureLoadv2, vecTuner: Provide good scaling and estimates.

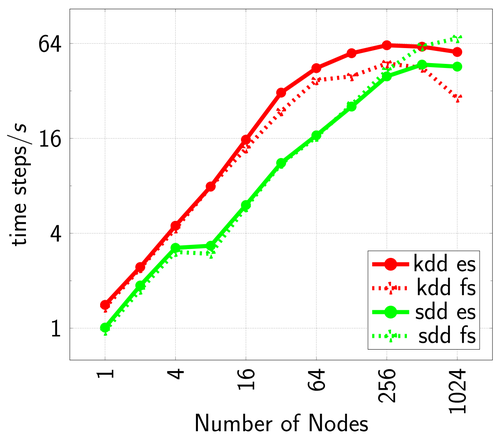


Figure 6: Influence of the Eighth Shell method to the sdd and kdd partitioning schemes for the 3M scenario with 8 OpenMP threads per rank. Uses the vectuner.

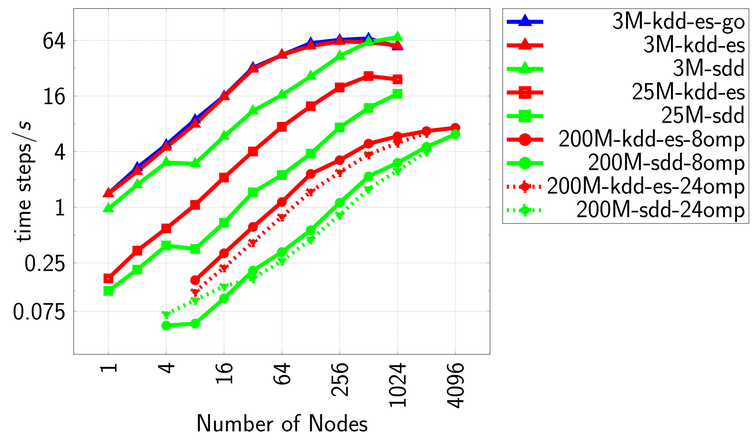


Figure 7: Comparison of the different scenarios for the measurement technique based on the vectorization tuner. If not indicated, 8 OpenMP threads are used per rank.

- Eighth shell method: speed-ups in the strong scaling limits for kdd, but not for sdd.
- More OpenMP threads beneficial for low node count. For larger node counts fewer threads should be used.
- Small benefits of global overlapping ("go") directives visible.
- k-d tree based load balancing now always performs better than standard domain decomposition, except for small scenarios + high node counts.

References

[1] K. J. Bowers, R. O. Dror, and D. E. Shaw.
Zonal methods for the parallel execution of range-limited n-body simulations.
Journal of Computational Physics, 221(1):303–329, 2007.

[2] N. Tchipev, S. Seckler, M. Heinen, J. Vrabec, F. Gratl, M. Horsch, M. Bernreuther, C. W. Glass, C. Niethammer, N. Hammer, B. Krischok, M. Resch, D. Kranzlmüller, H. Hasse, H.-J. Bungartz, and P. Neumann.
Twetris: Twenty trillion-atom simulation.
The International Journal of High Performance Computing Applications, 0(0):1094342018819741, 0.

Figure 8: https://github.com/ls1mardyn/ls1-mardyn/

Figure 9: https://github.com/AutoPas/AutoPas

www5.in.tum.de

steffen.seckler@tum.de