Task-Based Approaches for Molecular Dynamics Simulations

Fabio Gratl, Nikola Tchiepev, Steffen Seckler, Dr. Phillip Neumann, Prof. Dr. Hans-Joachim Bungartz
Technical University of Munich
Faculty of Informatics
Chair of Scientific Computing in Computer Science (SCCS)
Munich, 22.03.2018
Tools

ls1-mardyn

QuickSched

Fast Multipole Method in Tasks

Overview

Domain Structuring

FMM Steps

Results

Strong Scaling

Scheduling Analysis

Comparison to MPI Version
ls1-mardyn\textsuperscript{1} Overview

- Molecular dynamics simulation
- Language: C++
- Optimized for huge number of particles.
- Applications: nanofluids, nucleation, cavitation.
- Here: only single node level (shared memory).

\textsuperscript{1}http://www.ls1-mardyn.de

Source: Nikola Tchipev
QuickSched Overview

- Task based parallelism
- Language: C
- OpenMP or Pthreads
- Task dependencies
- Resource dependencies (Conflicts)
- Developers:
  Pedro Gonnet\(^2\)\(^3\), Aidan B.G. Chalk \(^2\), and Matthieu Schaller \(^4\)

→ **SWIFT** cosmological hydrodynamical code.

Download: [https://gitlab.cosma.dur.ac.uk/swift/quicksched](https://gitlab.cosma.dur.ac.uk/swift/quicksched)

\(^2\) School of Engineering and Computing Sciences, Durham University, United Kingdom.
\(^3\) Google Switzerland GmbH, Zürich, Switzerland.
\(^4\) Institute for Computational Cosmology, Durham University, United Kingdom.
Fast Multipole Method - Basic Ideas

- Split interaction calculation in near- and far-field.

\[ \Phi = \Phi_{\text{near}} + \Phi_{\text{far}} \]

- For near-field: use established efficient algorithms.
- For far-field: cluster groups of particles.
Domain Structuring

- Particle Cells with halo.
- FMM Cells tree, recursively subdividing the domain.
- FMM Cell contains one multipole and local expansion.
- Uniform Cells per level
FMM Flow of Information

- All incoming actions need to be done before outgoing can start.
- No two incoming actions can happen at the same time.
- Particle to Multipole (P2M).
- Multipole to Multipole (M2M).
- One Tasks includes $2^d$ sources and one target.
FMM Steps - Horizontal Pass

- Multipole to Local (M2L).
- One Task includes $6^d - 3^d$ sources and one target.
FMM Steps - Downward Pass

- Local to Local (L2L).
- Local to Particle (L2P).
- One Tasks includes one source and $2^d$ target.
FMM Steps - Near Field Evaluation

- Particle to Particle (P2P).
- Part of the Downward Pass.
Near Field Evaluation (P2P)

- Similar to Linked Cell (no cutoff).
- Use tightly packed pattern with Newton 3.

\[ F_{A,B} = -F_{B,A} \]

⇒ One task includes \(2^d\) cells.
Strong Scaling

Time to solution [s]

Speedup

Efficiency

- higher **Order** of multipole/local expansion
  - higher precision but longer time to solution.
- higher **Subdivision** factor (increases tree depth)
  - smaller particle cells, shift work from P2P to M2L.
Scheduling Analysis - Order

- Orders: Up 10, Down 31
- Lower x-axis almost three times longer.
- M2M ■ and L2L ■ grow faster than M2L ■
Scheduling Analysis - Subdivision Factor

- Subdivision Factors:
  Up 1, Down 2
- Lower x-axis about four times longer.
- M2L dominates.
Scheduling Analysis - Order + Subdivision Factor

- Orders: Up 10, Down 31
- Subdivision Factors:
  - Up 1, Down 2
  - Lower x-axis about 19 times longer.
- Many large tasks.
Comparison to MPI Version

Blue: *QuickSched*, orange: *MPI*
Conclusion

• Parallelizing FMM with QuickSched produces good results (scaling and time to solution).

• Comparable performance to MPI version.

• Choice of FMM parameters has significant effect on scalability.

⇒ Better task patterns might improve this.