Optimizing the Node-Level Performance of a Hydrodynamics Simulation Modeling Binary Star Systems on an Intel Knights Landing CPU

Gregor Daiß, David Pfander, Hartmut Kaiser

Louisiana State University

LSU Center for Computation & Technology

STE||AR GROUP

What is Octo-tiger?

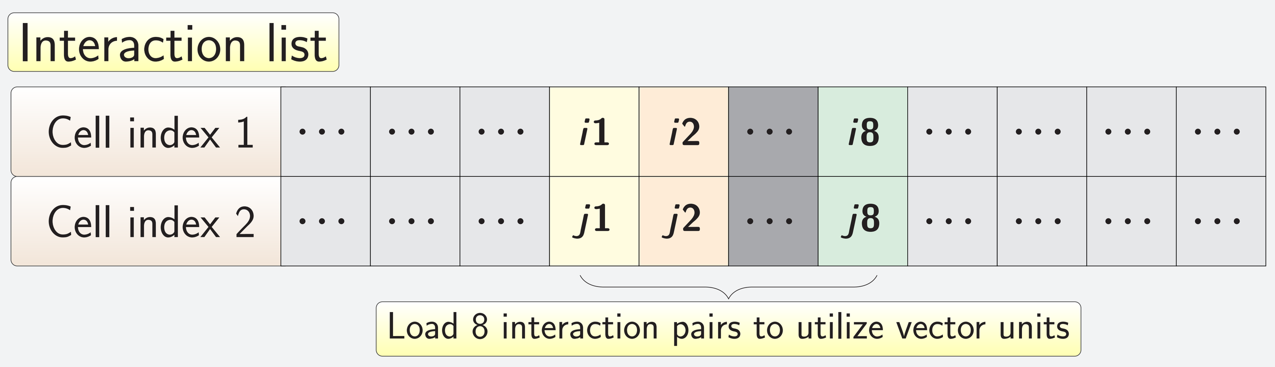
- Octo-tiger simulates self-gravitating astrophysical fluids. It is used in the modeling of interacting binary star systems
- To compute the gravitational field, it uses the fast multipole method
- Octo-tiger uses HPX to for parallelisation to run on distributed computing clusters

What are the Performance Bottlenecks?

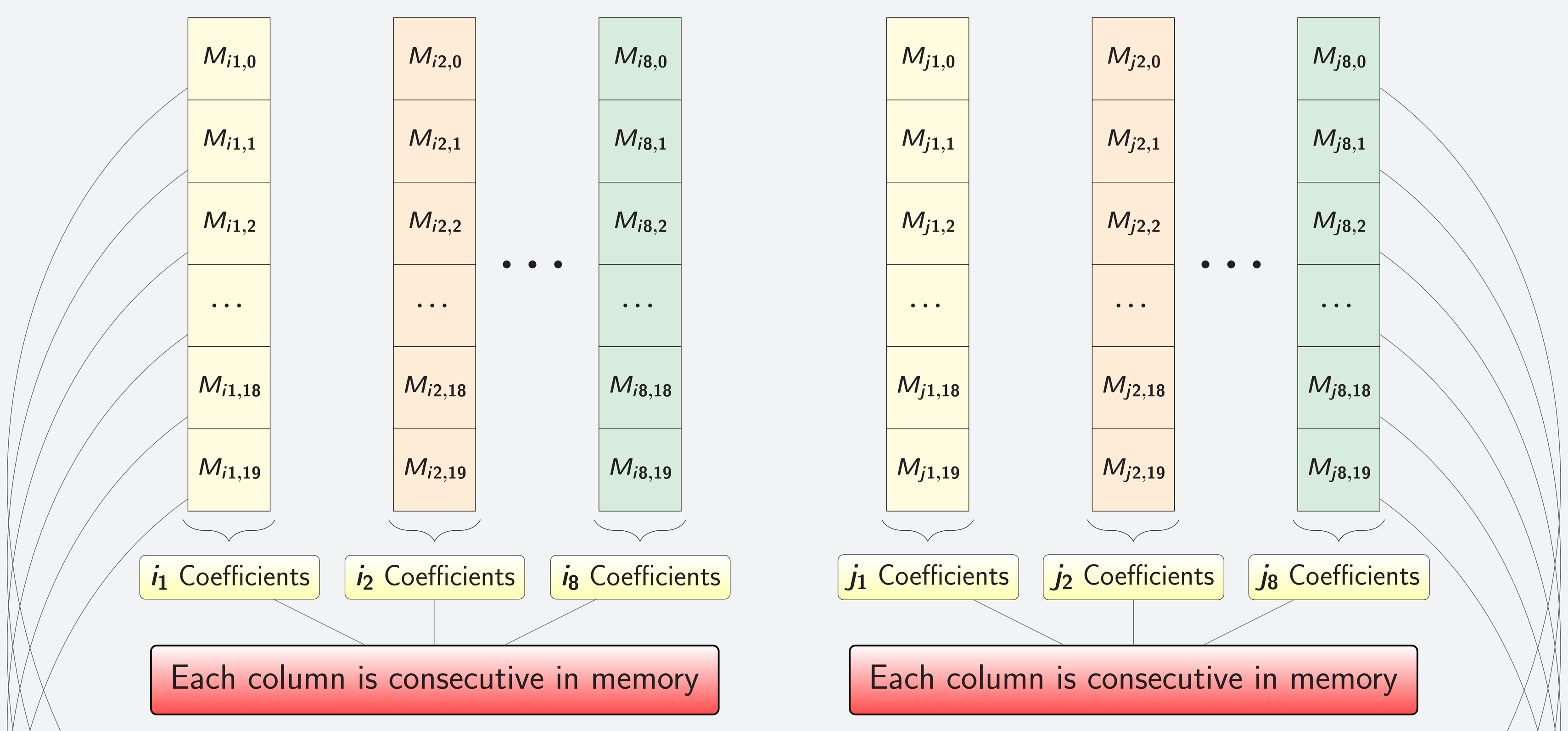
- Most time is spent calculating the gravitational field with the fast multipole method (FMM)
- Within the FMM methods, most time is spent computing the interactions of a cell with its close neighbors

A Closer Look at the old Interaction Calculation

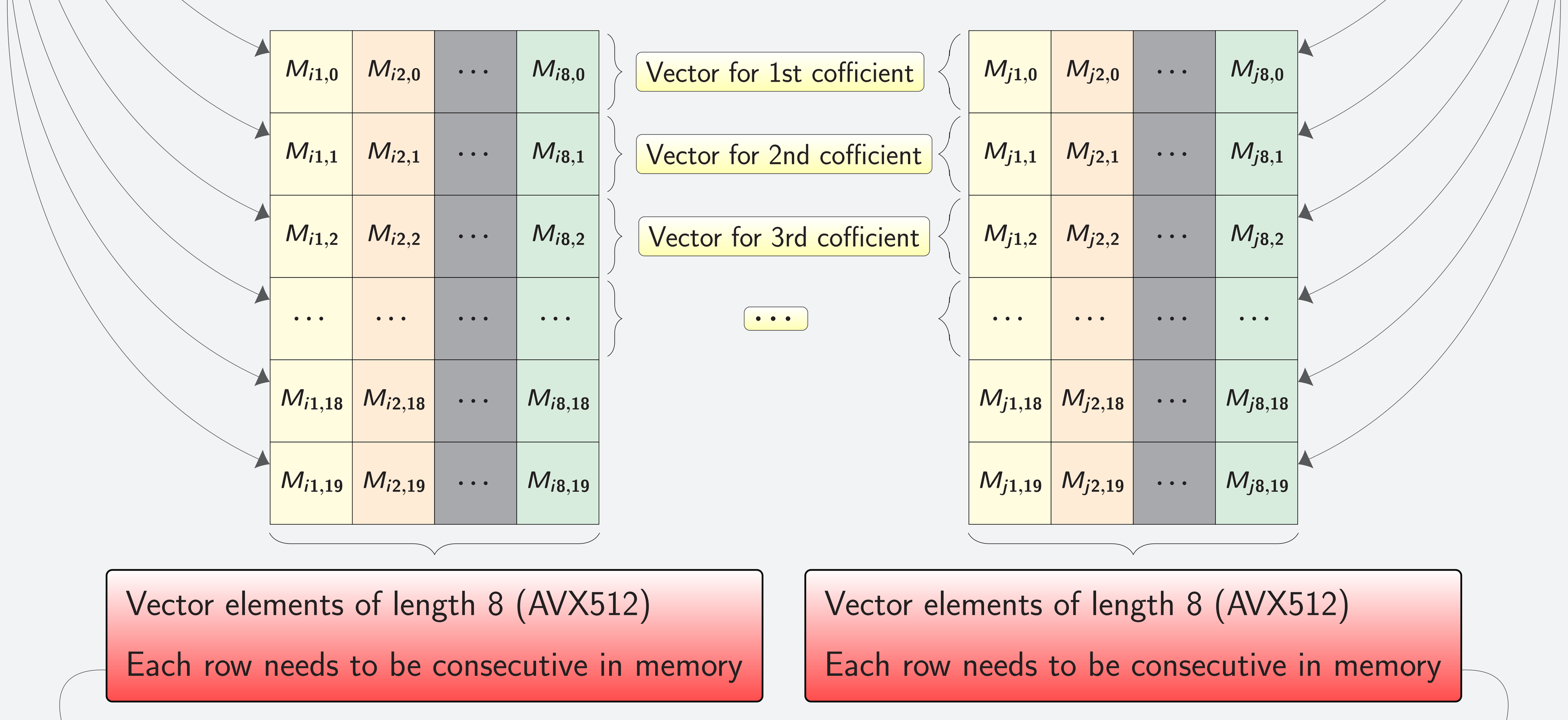
1. Find Interaction Pair i,j (load 8 at once)



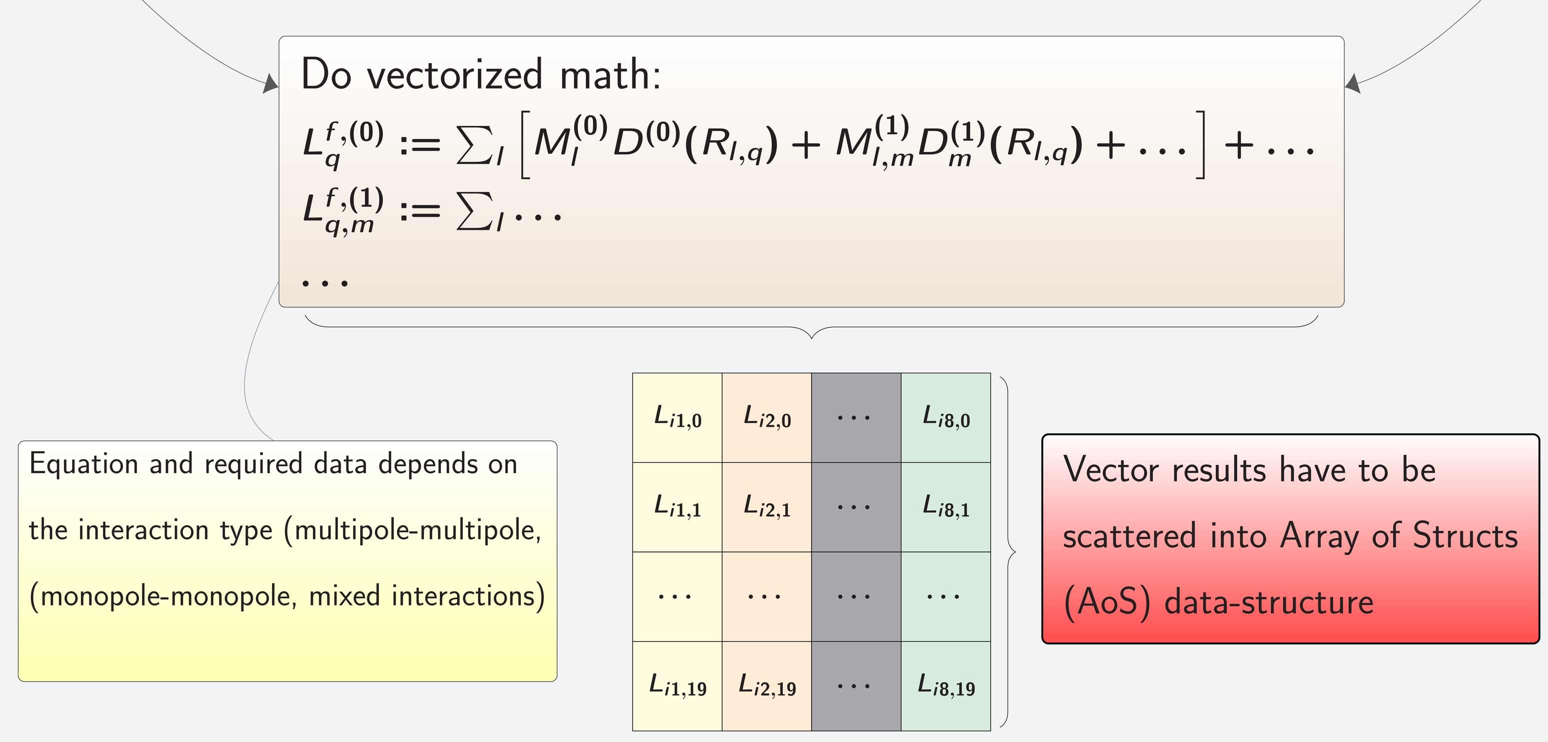
2. Load Coefficients of i,j (Array of Structs)



3. Data Alignment (into Vector Registers)



4. Calculate Interactions (and Store Results)



Insights

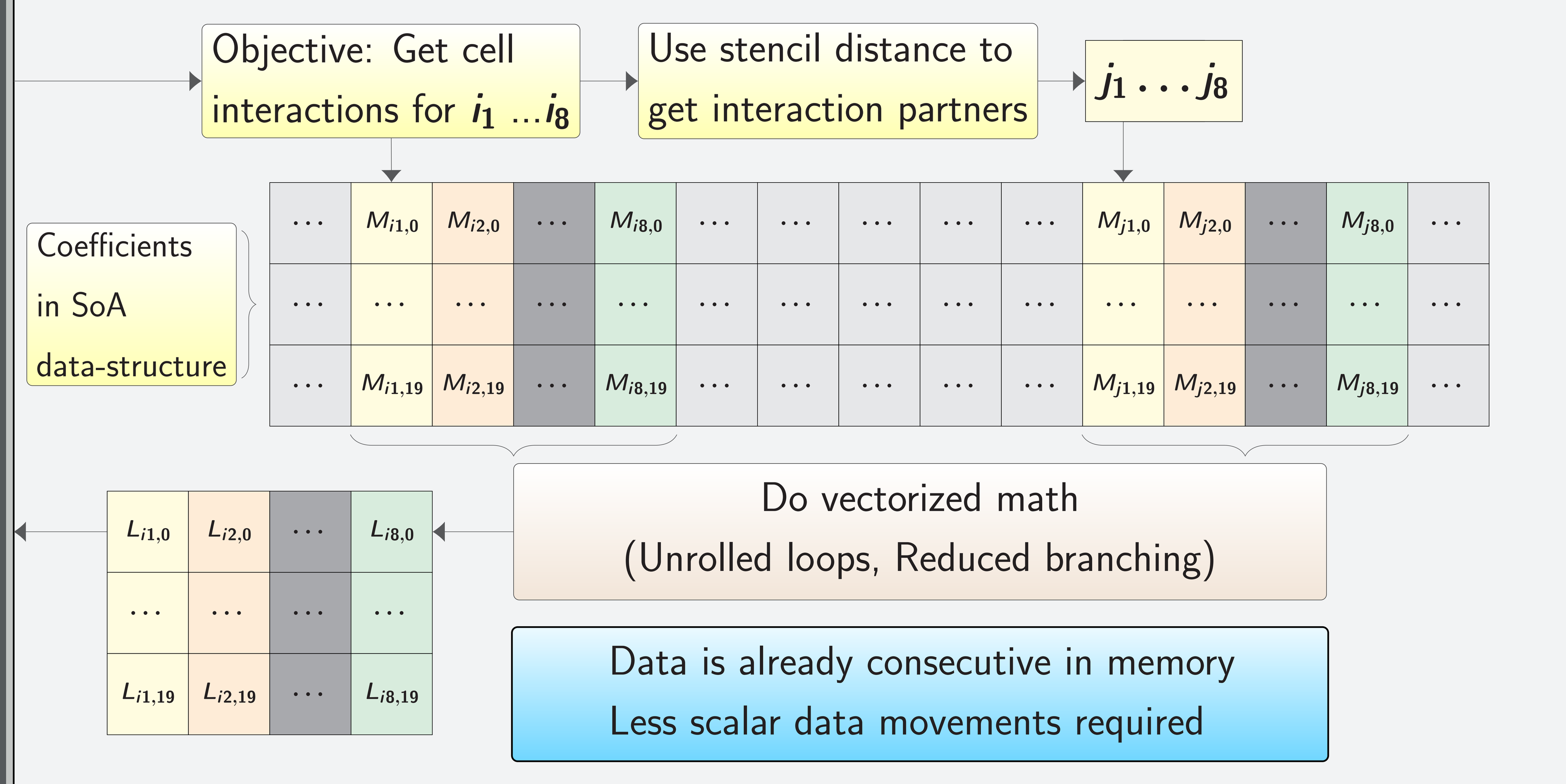
- The coefficient data of a cell is stored in a struct instance which is consecutive in memory. We cannot load a coefficient vector in one step
- Loading the coefficients of different cells into a vector register involves expensive gather operations
- We need to scatter the results from the vector register back into the structs

How to Improve the Performance

- Store coefficient data and results component-wise as Struct of Array (SoA)
  - Coefficients can be loaded directly into the vector registers
  - Interaction partners need to be next to each other in the memory
  - Therefore, use stencil approach instead of interaction list

New Interaction Method

Calculate Interactions for given Cells $i_1 \dots i_8$



Node-level Performance Comparison of both Interaction Methods

Platform (using one core only)	Old Version	New Version	Speedup
Intel Xeon Phi 7210 @ 1.30Ghz	21706.20s	6855.76s	3.17
Intel Xeon E5-2630 v3 @ 2.40GHz	1964.89s	1367.79s	1.44
Intel Xeon E5-2660 v3 @ 2.60GHz	1910.04s	1365.59s	1.40

Octo-tiger arguments for used scenario:
-t1 -Disableoutput -Problem=moving star -Max level=6 -Odt=0.3 -Stoptime=0.2 -Xscale=20.0 -Omega=0.1 -Stopstep=20

Conclusion

- By switching from an AoS data structure to an SoA one, we achieve a speedup of over 3.17 on KNL CPUs and a speedup of about 1.44 for Haswell CPUs.
- The new data structure is thus far only used for the calculations of the interactions. The speedup can be further improved by completely eliminating the old AoS data-structure and with that any conversions between the two structures

Acknowledgments

- Our thanks go to Dominic C. Marcello for all his explanations and answers, which helped us to better understand Octo-tiger and its structure!
- This material is based upon work supported by the National Science Foundation under Grant No. 1737785. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Dominic C. Marcello, Kundan Kadam, Geoffrey C. Clayton, Juan Frank, Hartmut Kaiser, and Patrick M. Motl.
Introducing octo-tiger/hpx: Simulating interacting binaries with adaptive mesh refinement and the fast multipole method.
Proceedings of Science, under review.