Algorithmic Patterns for H-matrices on Many-core Processors

Peter Zaspel
University of Basel
peter.zaspel@unibas.ch

Abstract

In this work, we consider the reformulation of hierarchical ($\mathcal{H}$) matrix algorithms for many-core processors with a model implementation on graphics processing units (GPUs). $\mathcal{H}$ matrices approximate specific dense matrices, e.g., from discretized integral equations or kernel ridge regression, leading to log-linear time complexity in dense matrix-vector products. The parallelization of $\mathcal{H}$ matrix operations on many-core processors is difficult due to the complex nature of the underlying algorithms. While previous algorithmic advances for many-core hardware focused on accelerating existing $\mathcal{H}$ matrix CPU implementations by many-core processors, we here aim at totally relying on that processor type. As main contribution of the presentation, the necessary parallel algorithmic patterns allowing to map the full $\mathcal{H}$ matrix construction and the fast matrix-vector product to many-core hardware are introduced. Here, crucial ingredients are space filling curves, parallel tree traversal and batching of linear algebra operations. The resulting model GPU implementation `hmglib` is the, to the best of the authors knowledge, first entirely GPU-based Open Source $\mathcal{H}$ matrix library of this kind. The presentation is concluded by an in-depth performance analysis and a comparative performance study against a standard $\mathcal{H}$ matrix library, highlighting profound speedups of our many-core parallel approach.

References

1. P. Zaspel. Algorithmic patterns for H-matrices on many-core processors. arXiv preprint, arXiv:1708.09707.