Accelerating Tensor Contractions for High-Order FEM on CPUs, GPUs, and KNLS

Azzam Haidar, Ahmad Abdelfattah, Veselin Dobrev, Ian Karlin, Tzanio Kolev, Stanimire Tomov, and Jack Dongarra

Abstract
High-performance is difficult to obtain using existing libraries, especially for many independent computations where each computation is very small. However, using our framework to batch computation plus application-specifics, we demonstrate close to peak performance results. In particular, to accelerate large-scale tensor-formulated high-order finite element method (FEM) simulations, which is the main focus and motivation for this work, we represent contractions as tensor index reordering plus matrix-matrix multiplications (GEMMs). This is a key factor to achieve algorithmically many-fold acceleration (vs. not using it) due to possible reuse of data loaded in fast memory.

Motivation
Numerous important applications can be expressed through tensors:
- High-order FEM simulations
- Signal Processing
- Numerical Linear Algebra
- Numerical Analysis

Accelerating High-order FEM
Lagrangian Hydrodynamics in the BLAST code:

Methodology, Design, and Optimization
We conducted an extensive study over the performance counters using the PAPI tools to conclude that in order to achieve an efficient execution, we need to maximize the occupancy and minimize the data traffic while respecting the underlying hierarchical memory design. Unfortunately, today’s compilers cannot introduce highly sophisticated cache/register based loop transformations.

- Data Access Optimizations and Loop Transformation Techniques
- Register Data Reuse and Locality
- Effect of the Multi-threading

Conclusions and Future directions
- High-performance package on Tensor Algebra has the potential for high-impact on a number of important applications
- Multidisiplinary effort
- Current results show promising performance, where various components will be leveraged from autotuning MAGMA Batched linear algebra kernels, and BLAST from LLNL.