ICL

MATEDOR

MATRIX, TENSOR, AND DEEP-LEARNING OPTIMIZED ROUTINES

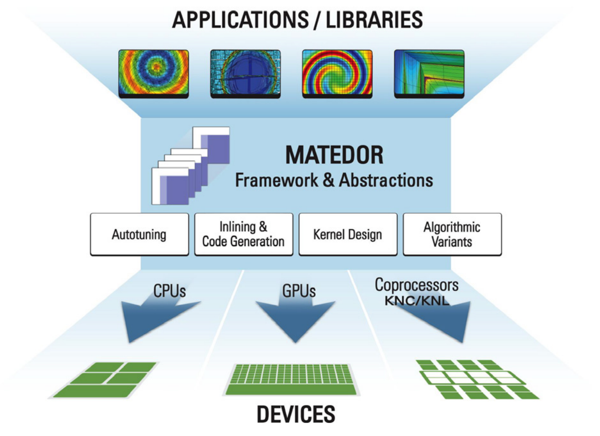
The MAtrix, TEnsor, and Deep-learning Optimized Routines (MATEDOR) project seeks to develop software technologies and standard APIs, along with a sustainable and portable library for large-scale computations, but whose individual parts are very small matrix or tensor computations. The main target is the acceleration of applications from important fields that fit this profile, including deep learning, data mining, astrophysics, image and signal processing, hydrodynamics, and more.

STANDARD INTERFACE FOR BATCHED ROUTINES

Working closely with affected application communities, we will define modular, language-agnostic interfaces that can be implemented to work seamlessly with the compiler and be optimizable using techniques such as code replacement and inlining. This will provide the developers of applications, compilers, and runtime systems with the option of expressing as a single call to a routine from the new batch operation standard, and would allow the entire linear algebra (LA) community to collectively attack a wide range of small matrix or tensor problems. Success in such an effort will require innovations in interface design, computational and numerical optimization, as well as packaging and deployment at the user site to trigger final stages of tuning at the moment of execution.

SUSTAINABLE AND PERFORMANCE-PORTABLE SOFTWARE LIBRARY

We will demonstrate the power of the MATEDOR interface by delivering a high-performance numerical library for batched LA subroutines autotuned for the modern processor architecture and system designs. The MATEDOR library will include LAPACK routine equivalents for many small dense problems, tensor, and application-specific operations, e.g., for deep learning; these routines will be constructed as much as possible out of calls to batched BLAS routines and their look-alikes required in sparse computation context.



APPLICATIONS / LIBRARIES

MATEDOR

Framework & Abstractions

Autotuning

Inlining & Code Generation

Kernel Design

Algorithmic Variants

CPUs

GPUs

Coprocessors KNC/KNL

DEVICES

ENABLING TECHNOLOGIES

MATEDOR will develop enabling technologies for very small matrix and tensor computations, including: (1) autotuning, (2) inligning, (3) code generation, and (4) algorithmic variants. We define the success of the research conducted and the software developed under the MATEDOR project as being able to automate these four aspects to allow for both flexibility and close-to-optimal performance of the final code that gets used by the domain scientist.

STANDARD APIS (FOR BATCHED BLAS AND LAPACK)

Proposed API is very similar to the standard BLAS/LAPACK API

```
void dgemm_batched (
    batched_trans_t transA , batched_trans_t transB ,
    batched_int_t m, batched_int_t n, batched_int_t k,
    double alpha ,
    double const * const * dA_array , batched_int_t ldda ,
    double const * const * dB_array , batched_int_t lddb ,
    double beta ,
    double ** dC_array , batched_int_t lddc ,
    batched_int_t batchCount , batched_queue_t queue
    batched_int_t *info );
```

PUBLICATIONS

A. Abdelfattah, A. Haidar, S. Tomov, and J. Dongarra, **"Tensor Contractions using Optimized Batch GEMM Routines,"** March 26-29 2018, GPU Technology Conference (GTC), Poster, San Jose, CA. [Online]. Available: http://icl.cs.utk.edu/magma/software/

L. Ng, K. Wong, A. Haidar, S. Tomov, and J. Dongarra, **"MagmaDNN High-Performance Data Analytics for Manycore GPUs and CPUs,"** December 2017, MagmaDNN, 2017 Summer Research Experiences for Undergraduate (REU), Knoxville, TN. [Online]. Available: http://icl.cs.utk.edu/magma/software/

A. Haidar, A. Abdelfattah, M. Zounon, S. Tomov, and J. Dongarra, **"A Guide For Achieving High Performance With Very Small Matrices On GPU: A case Study of Batched LU and Cholesky Factorizations,"** IEEE Transactions on Parallel and Distributed Systems, vol. PP, no. 99, pp. 1–1, 2017.

A. Abdelfattah, A. Haidar, S. Tomov, and J. Dongarra, **"Batched one-sided factorizations of tiny matrices using GPUs: Challenges and countermeasures,"** Journal of Computational Science, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877750317311456

T. Dong, A. Haidar, S. Tomov, and J. Dongarra, **"Accelerating the SVD bi-diagonalization of a batch of small matrices using GPUs,"** Journal of Computational Science, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187775031731150X

I. Yamazaki, A. Abdelfattah, A. Ida, S. Ohshima. S. Tomov, R. Yokota, J. Dongarra, **"Performance of Hierarchical-matrix BiCGStab Solver on GPU clusters,"** 2018 IEEE International Parallel & Distributed Processing Symposium.

WITH SUPPORT FROM NSF

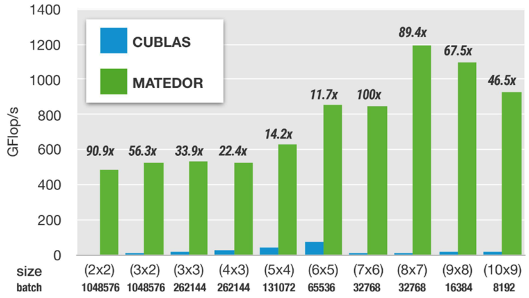
ICL

MATEDOR

BREADTH OF MATEDOR'S IMPACT ON APPLICATION DOMAINS

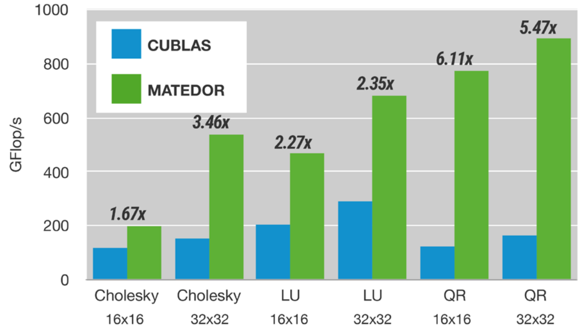
TENSOR CONTRACTIONS IN HIGH ORDER FEM & APPLICATIONS

Tensor Contractions: computing $B^T D (BAB^T) B$,
double precision, Tesla V100 GPU



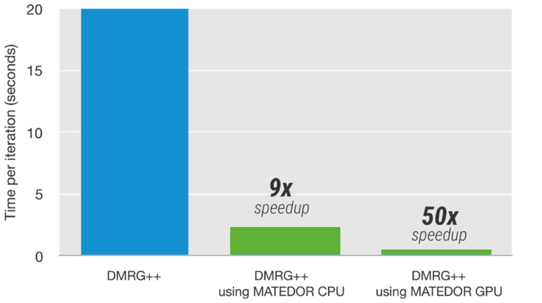
SPARSE/DENSE SOLVERS & PRECONDITIONERS

Batch Matrix Factorization, 100k matrices,
double precision, Tesla V100 GPU



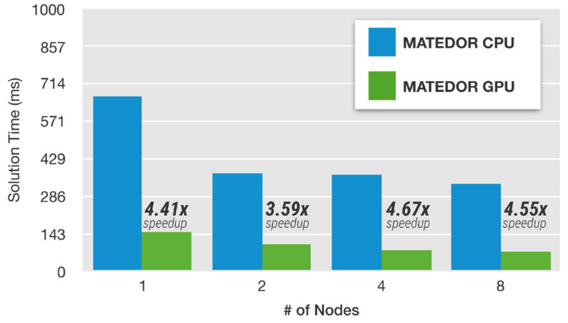
HIERARCHICAL LINEAR SOLVERS ON GPU CLUSTERS

DRMG++ Acceleration using MATEDOR Batched computations

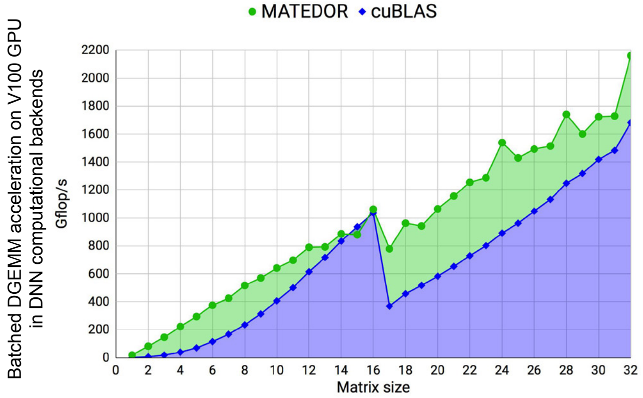


DENSITY MATRIX RENORMALIZATION GROUP DMRG++

Hierarchical Linear Solver, 2 P100 GPUs per node



DEEP NEURAL NETWORKS AND DATA ANALYTICS



WITH SUPPORT FROM NSF

This work is partially supported by NSF Grant No. OAC 1740250 and CSR 1514286, NVIDIA, and the Department of Energy under the Exascale Computing Project (17-SC-20-SC and LLNL subcontract under DOE contract DE-AC52-07NA27344).