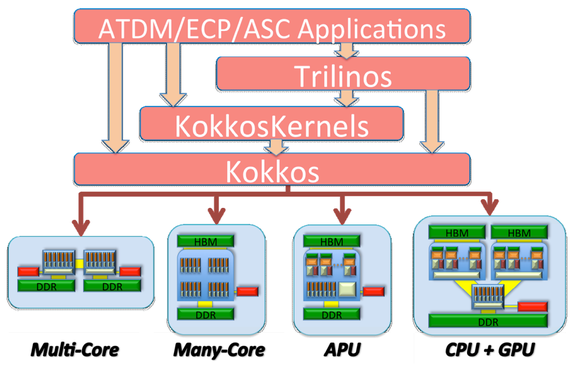
KokkosKernels: Performance-Portable Sparse, Dense and Graph Kernels

SAND2017-13015D

Sandia National Laboratories
Andrew Bradley, Mehmet Deveci, Simon Hammond, Mark Hoemmen, Kyungjoo Kim, Sivasankaran Rajamanickam, Christian Trott
Center for Computing Research, Sandia National Laboratories, NM, USA

Overview

- Layer of performance-portable kernels
- Provide kernels usable within different levels of parallelism (wherever applicable): *Device level, Team level, Thread level, Serial*
- No dependencies other than Kokkos
- Node-level Kernels (No MPI)
- Can utilize multiple memory-spaces (wherever applicable)



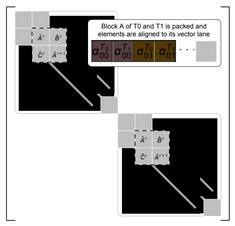
Capabilities

Concept	Example
Sparse Linear Algebra	Sparse Matrix-Vector Multiply, Sparse Matrix-Matrix Multiply, Gauss-Seidel
Dense Linear Algebra	BLAS1 and BLAS2 kernels, Batched BLAS operations for LU factorizations (LU), dense matrix-matrix multiply (GEMM) and dense triangular solver (TRSM)
Graph Algorithms	Distance-1 Graph Coloring
Data Structures	Hashmap - Allows thread scalable accumulations based on keys Uniform Memory Pool – Thread scalable way to request and use memory from a persistent memory space
Upcoming	Distance-2 Graph Coloring, Contraction Kernels, Other batched operations

Sibling projects

Kokkos	Programming Model for Performance Portability
ShyLU	Kokkos based solvers for direct/fine-grained factorizations and triangular solves on the node

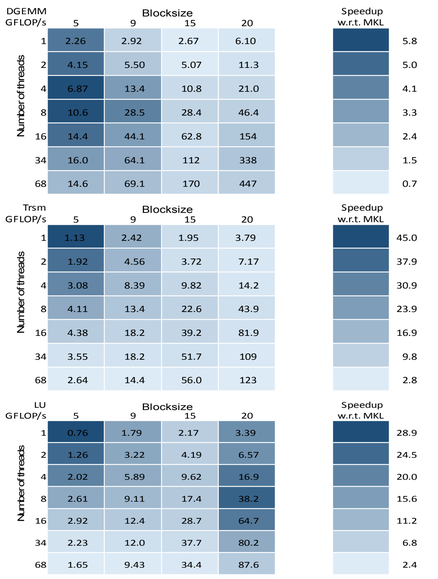
Batched Dense Linear Algebra (LU, GEMM, TRSM)



Algorithm 2: Batched impl. TriLU

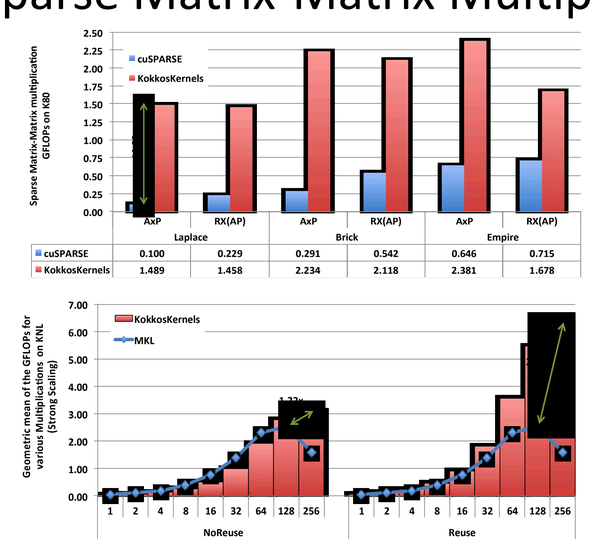
1 for a pair $T(0,1)$ in $\{\{T_0, T_1\}, \{T_2, T_3\}, \cdots, \{T_{m\_n-2}, T_{m\_n-1}\}\}$ do in parallel
2 for r 0 to $k-2$ do
3 $\hat{A}^{r(0,1)} := LU(\hat{A}^{r(0,1)})$;
4 $\hat{B}^{r(0,1)} := L^{-1}\hat{B}^{r(0,1)}$;
5 $\hat{C}^{r(0,1)} := \hat{C}^{r(0,1)}U^{-1}$;
6 $\hat{A}^{r+1(0,1)} := \hat{C}^{r+1(0,1)} - \hat{C}^{r(0,1)}\hat{B}^{r(0,1)}$;
7 end
8 $\hat{A}^{k-1(0,1)} := \{L \cdot U\}$;
9 end

- Batched dense linear algebra kernels need to be sensitive to the hierarchical parallelism
- Threaded Block Tridiagonal factorization motivated by line preconditioners for multiphysics applications. Small physics blocks motivate vector level batched BLAS
- Packed, vectorized, batched BLAS calls for GEMM, TRSM, LU results in speed up of up to 5.8x, 45x, 28.9x, respectively.



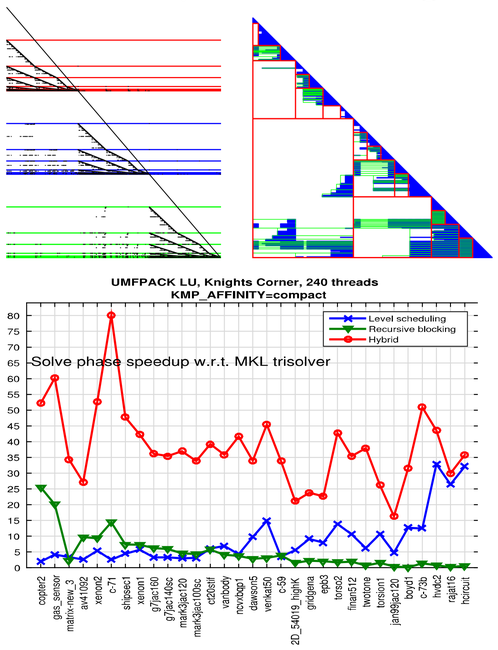
KNL comparisons w/ batched DGEMM, unbatched LU & TRSM. GFOLP/s (numbers) and speedup w.r.t MKL (colors) shown for 512 worksets

Sparse Matrix-Matrix Multiplication (SpGEMM)



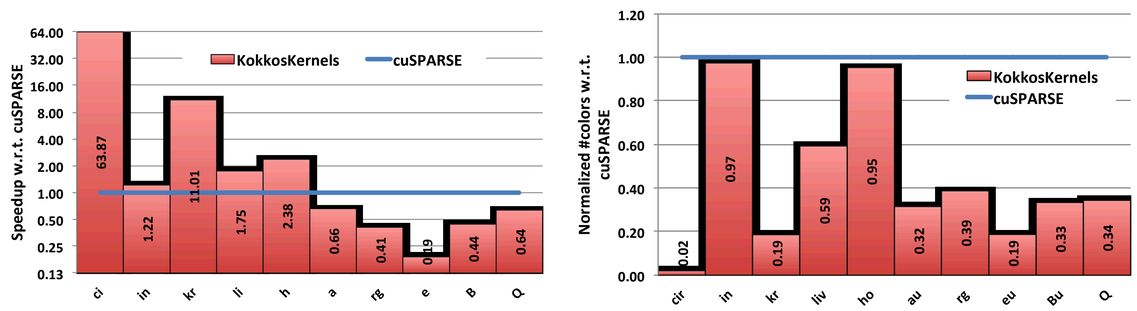
- SpGEMM is the most expensive part of the multigrid setup.
- New portable write-avoiding algorithm in KokkosKernels is ~14x faster than NVIDIA's CUSPARSE on K80 GPUs.
- ~2.8x faster than Intel's MKL on Intel's Knights Landing (KNL).
- Memory scalable: Solving larger problems that cannot be solved by codes like NVIDIA's CUSP and Intel's MKL when using large number of threads.

Hybrid Sparse Triangular Solver (ShyLU/HTS)

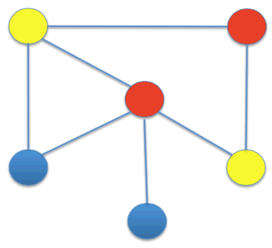


- Sparse Triangular Solve is the key kernel in several preconditioners (BDDC, GDSW)
- Hybrid Sparse Triangular Solver – Uses combination of level-sets and recursive blocking
- P2P synchronization between threads utilized for better performance
- OpenMP based implementation: Speedups ranging from 15x-80x Intel KNC, and 7x-17x on Ivybridge
- Needs ~8 parallel triangular solves to amortize the cost of reloading the data

Graph Coloring and Multi-threaded Gauss-Seidel



- Goal: Identify independent data that can be processed in parallel.
- Performance: Better quality (4x on average) and run time (1.5x speedup) w.r.t cuSPARSE.
- Enables parallelization of preconditioners: Gauss Seidel: 82x speedup on KNC, 136x on K20 GPUs



Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.