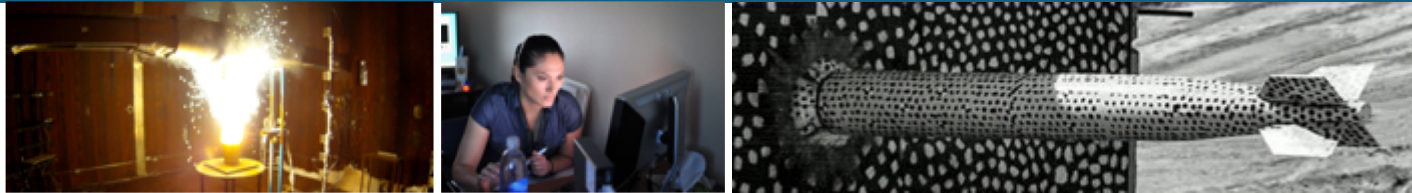


Batched Reproducible and Reduced Precision BLAS Forum – SC'18



PRESENTED BY

Siva Rajamanickam

Kyungjoo Kim, Vinh Dang, Andrew Bradley, Micah Howard, Sandia National Laboratories

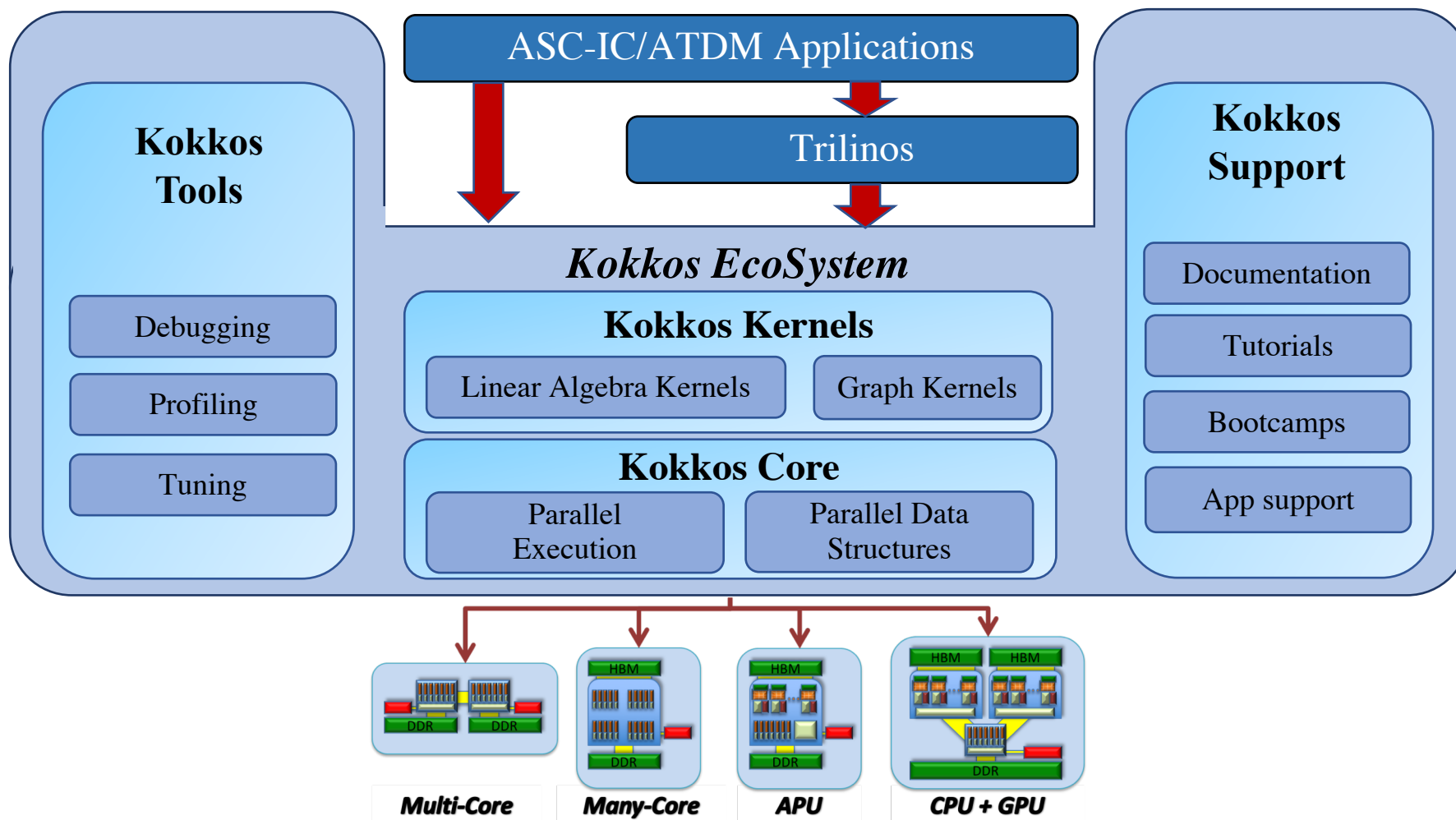


Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.



Compact BLAS in Kokkos Kernels update



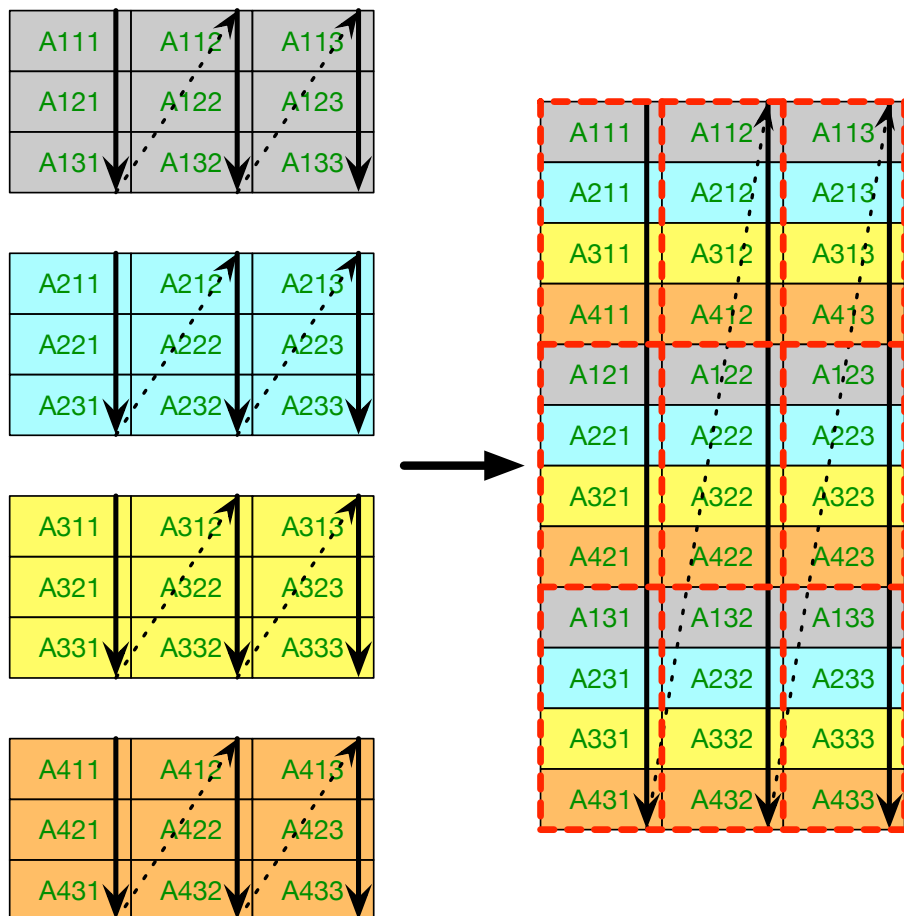


Kokkos Core: parallel patterns and data structures, supports several execution and memory spaces

Kokkos Kernels: performance portable BLAS, sparse, and graph algorithms and kernels

Kokkos Tools: debugging and profiling support

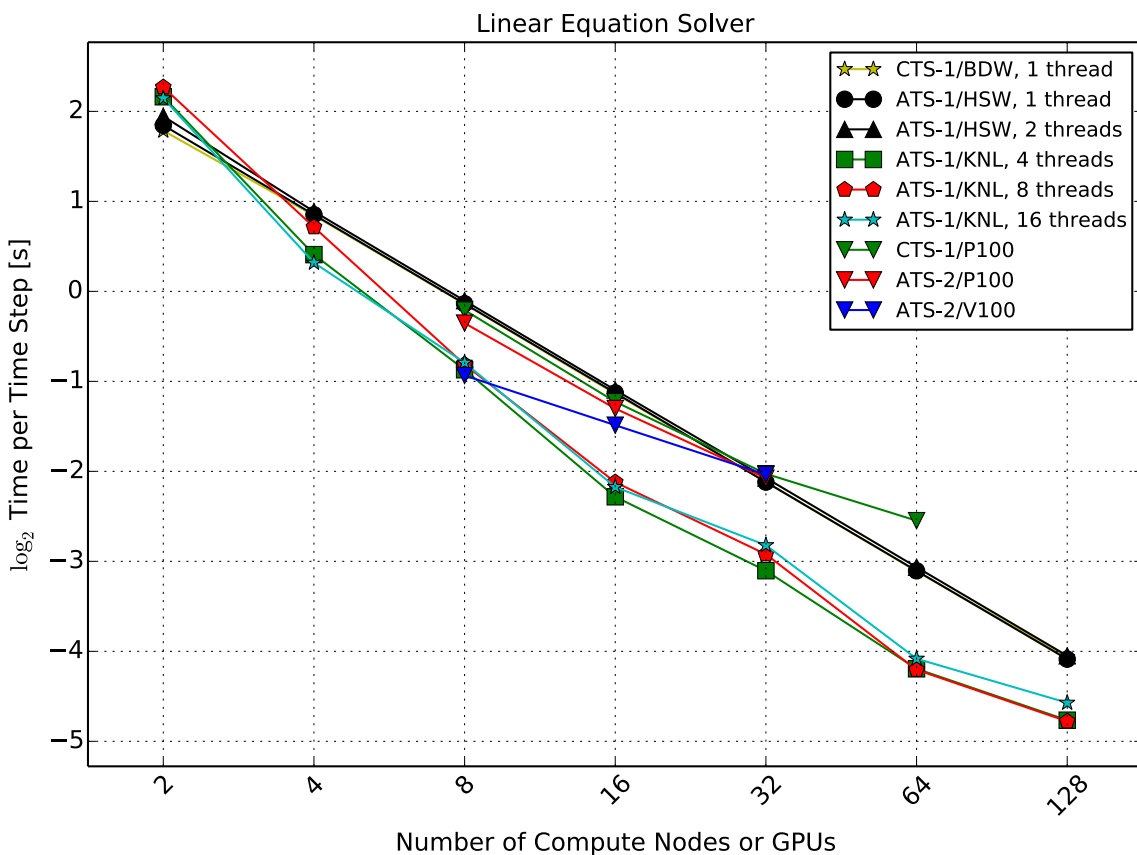
Kokkos Ecosystem addresses complexity of supporting numerous many/multi-core architectures that are central to new Supercomputers



Standard data layout

Compact data layout
using vector length of 4

- Data layouts are critical for performance
 - Divide the batch of matrices in blocks
 - Interleave the blocks
 - Block enough to fill vector units and fit in cache
- Compact layout: Block size is same as vector length
- Pros for compact layout: Improves vectorization
- Cons for compact layout: Might need repacking the data for new layouts
- See SC'17 paper for performance comparisons
 - K. Kim, T.B. Costa, M. Deveci, A.M. Bradley, S.D. Hammond, M.E. Guney, S. Knepper, S. Story, S. Rajamanickam, "Designing Vector-Friendly Compact BLAS and LAPACK Kernels," SC17.



- Developed new preconditioners based on Compact BLAS Kernels
- Integrated the compact BLAS based preconditioners into production application (SPARC) that is part of the Exascale Computing Program and ATDM program
 - See ICCFD paper: *Howard, M., T. Fisher, M. Hoemmen, D. Dinzl, J. Overfelt, A. Bradley, K. Kim, and S. Rajamanickam. "Employing Multiple Levels of Parallelism for CFD at Large Scales on Next Generation High-Performance Computing Platforms."*
- Continued work with Intel Compact BLAS
- Home grown version of Compact BLAS on the GPUs as required by the applications
 - Performance can be improved with vendor support (hint, hint ..)*
- Support for several new kernels getri, getsr,
- Working on other new kernel QR, Hessenberg form, Eigen solvers



- Continue to develop team level compact kernels
 - Contrary to Batched BLAS community's original plans most popular application need is the compact BLAS over traditional way of calling batched BLAS
 - Applications like control of “parallel for” and call BLAS within the loop (teams)
 - Use cases different – minimal error checking, vectorization needs high
 - Use cases for variable block size growing
 - <https://github.com/kokkos/kokkos-kernels/issues/9>
- Some of the above use cases are addressed in some implementations
 - No complete set
 - GPU support for compact BLAS is lacking
- Standardization of the C++ version of the reference, with team level interface, would be nice
- One document for C, C++, team level and device level interface would be nice