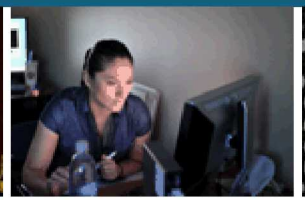
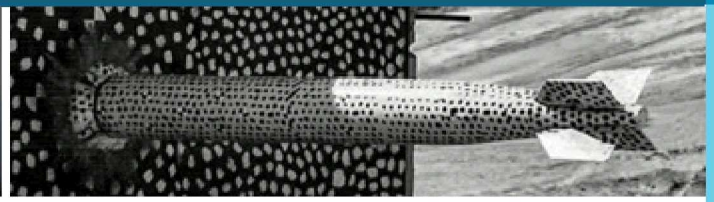
SAND2019-4539C

Kokkos Kernels





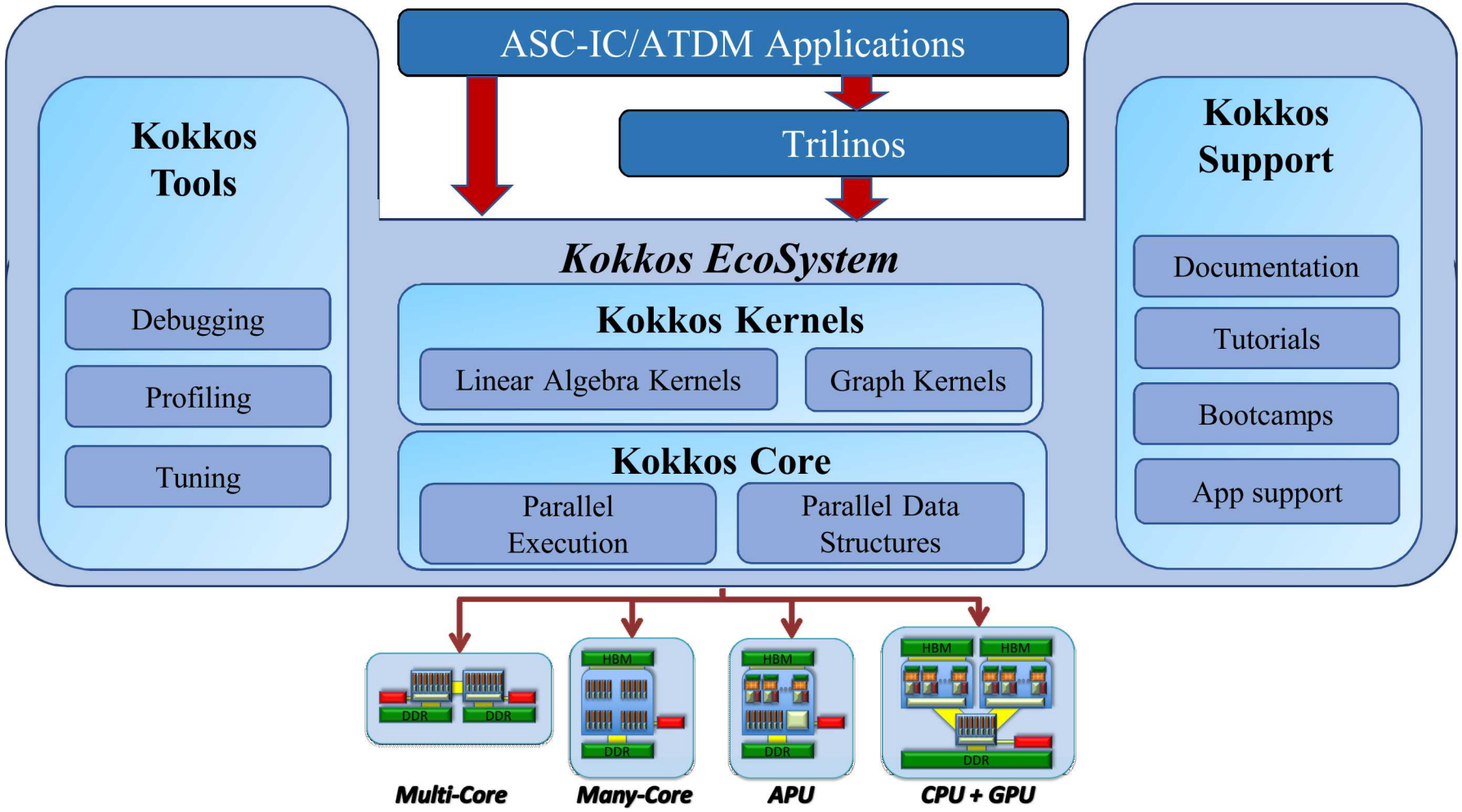
PRESENTED BY

Siva Rajamanickam, Luc Berger-Vergiat, Vinh Dang, Nathan Ellingwood, Kyungjoo Kim, Will McLendon, Christian Trott, Jeremiah Wilke

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.

SAND2019-XXXXXX

Kokkos Ecosystem for Performance Portability



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

Kokkos Kernels: performance portable BLAS; sparse, dense and graph algorithms

Kokkos Tools: debugging and profiling support

Write-once using Kokkos for portable performance on different architectures

Kokkos Ecosystem addresses complexity of supporting numerous many/multi-core architectures that are central to DOE HPC enterprise

Focus of Kokkos Kernels

Deliver ***portable*** sparse/dense linear algebra and graph kernels

- These are the kernels that are in 80% of time for most applications
- Key problems: Kernels might need different algorithms/implementations to get the best performance
- Ninja programming needs in addition to Kokkos
- Users of the kernels do not need to be ninja programmers
- ***Focus on performance of the kernels on all the platforms of interest to DOE***

Deliver ***robust software ecosystem*** for other software frameworks and applications

- Production software capabilities that give high performance, portable and turn-key
- Tested on number of configurations nightly (architectures, compilers, debug/optimized, programming model backend, complex/real, ordinal types…)
- Larger release/integration testing with Trilinos and applications
- Kokkos Support, github issues, tutorials, hackathons, user group meetings

Kokkos Kernels delivers portable, high-performance kernels in a robust software ecosystem

Primary Application and Program Drivers

Application Drivers

- SPARC: state-of-the-art hypersonic unsteady hybrid structured/unstructured finite volume CFD code
  - High performance line solvers; batched BLAS on CPUs and GPUs
  - Performance-portable programming models
- EMPIRE: next-gen unstructured-mesh FEM PIC/multifluid plasma simulation code
  - Thread-scalable, performance-portable, on-node linear algebra kernels to support multigrid methods
  - Performance-portable programming models
- Exawind: next-gen wind simulation code
  - Thread-scalable, performance-portable, on-node linear algebra kernels to support multigrid methods
  - Performance-portable programming models

CoDesign Drivers

- ExaLearn : Addressing Machine Learning needs of ECP applications
  - Batched Linear Algebra
  - High performance convolutions
- ExaGraph: Combinatorial Algorithms of ECP applications
  - Distance-1/Distance-2 Graph Coloring

SciDAC applications (CSP) and SciDAC projects (FASTMath): Batched linear algebra

Network Science or Data Science applications: Linear algebra based graph algorithms

ECP CLOVER: starting FY20

Released Features

Sparse Linear Algebra

- Sparse Matrix Vector Multiplication
- Sparse Matrix-Matrix Multiplication
- Gauss-Seidel Preconditioner

Dense Linear Algebra

- Batched linear algebra kernels, especially LU, GEMM, and TRSM, for small block matrices optimized on GPUs (K. Kim) *Kyungjoo will talk next*
- Interface BLAS calls from vendors and hand-rolled implementations

Graph Algorithms

- Distance-1 coloring algorithms for aggregation in algebraic methods

Kokkos Kernels has several new features based on application and codesign drivers

New Features in 2018/2019

Sparse Linear Algebra

- Sparse Matrix Vector multiplication for structured matrices (L. Berger-Vergiat)
- Level Set Triangular Solves (coming in 3.0, N. Ellingwood)

Dense Linear Algebra

- Batched linear algebra kernels, especially LU, GEMM, and TRSM, for small block matrices optimized on GPUs (K. Kim)
- Batched linear algebra kernels, especially eigen solvers, for medium-size block matrices (K. Kim)
- Batched linear algebra kernels for complex matrices (V. Dang)
- MAGMA support for complex dense matrices (V. Dang)

Graph Algorithms

- Distance-2 coloring algorithms for aggregation in algebraic methods (W. McLendon)

Kokkos Kernels has several new features based on application and codesign drivers

New Features in 2018/2019

Network Science Problems

- IEEE/Amazon/DARPA Graph Challenge "Champion" 2017
- IEEE/Amazon/DARPA Graph Challenge "Champion" 2018
- Fast Triangle counting using Kokkos/Cilk
  - M. Wolf, M.Deveci, A. Yasar, J. Berry. S. Hammond, and S. Rajamanickam
  - Cilk backend coming soon

Software

- One release process as a Kokkos ecosystem (N. Ellingwood)
- Lots more testing, examples, basic tutorial materials
- CMake support (almost there – in a branch) (J. Wilke)

Kokkos Kernels has several new features based on application and codesign drivers

FUTURE DIRECTIONS/ GOALS

Portable SIMD Type (coming soon)

Design Points

- Requirements to work on CPUs/traditional VPUs and SIMT style architectures
- Need support for logical vector lengths that are different from physical vector length
- No overhead costs as it is in the innermost loops

Preliminary implementation for KNL, ARM CPUs, and GPUs.

- Damodar Sahasrabudhe (University of Utah), with Eric Phipps, Siva Rajamanickam and Martin Berzins

Evaluated on batched linear algebra kernels, sparse matrix-multivector, and assembly kernels

- Up to 7.8xx, 2.2x, and 1.13x speed-up for the assembly kernel on KNL, ARM and GPUs respectively

Requires use of Portable temporary variable to be able to write to temporaries in VPUs and GPUs

Kokkos SIMD Type focuses on portability between VPUs and SIMT architectures

Future Directions (New Features)

Support machine learning needs of ECP applications
- Batched linear algebra
- Portable convolution kernels

Support new preconditioners and linear algebra kernels
- Address needs for robust preconditioners that are portable on GPUs (ILU)
- Support for new linear algebra kernels (, tensor contractions)

Support a SIMD data type

Support new graph kernels for analytics applications
- Linear algebra based graph kernels

Future Directions (Software)

New delivery mechanisms

- Spack support (CMake support almost there)
- Integrated into open source machine learning frameworks

Tutorials, Support of the product

- New architecture and backend support

Integration into more ECP applications

Support for experimental architectures for graph analytics

- Cilk backend
- Graph algorithms on Emu architecture