



Sandia
National
Laboratories

Exceptional service in the national interest

KOKKOS KERNELS

Luc Berger-Vergiat, Siva Rajamanickam
V. Dang, N. Ellingwood, J. Foucar, B. Kelley, E. Harvey,
K. Liegeois, C. Pearson, E. Prudencio

Trilinos User Group meeting 2023
Albuquerque, New Mexico



ECP / SAKE UPDATES

SAKE ACCOMPLISHMENTS

- KPP-3 integrations:
 - ATDM application integrations SPARC and EMPIRE
 - Focused on features for solvers and preconditioners on AMD platforms (BlockTriDiag, ILU, Multigrid...)
 - Supports application milestones
 - ECP integrations
 - Integration with Trilinos and PETSc
 - Integration on AMD and Intel platforms
- ALL KPP-3 are reviewed and approved by federal program manager
 - Trilinos contributes 1pt to Math Libraries
 - Kokkos Kernels contribute 0.5pt to Math Libraries, 0.5pt to ATDM
 - Sake amongst the first math libraries project fully approved



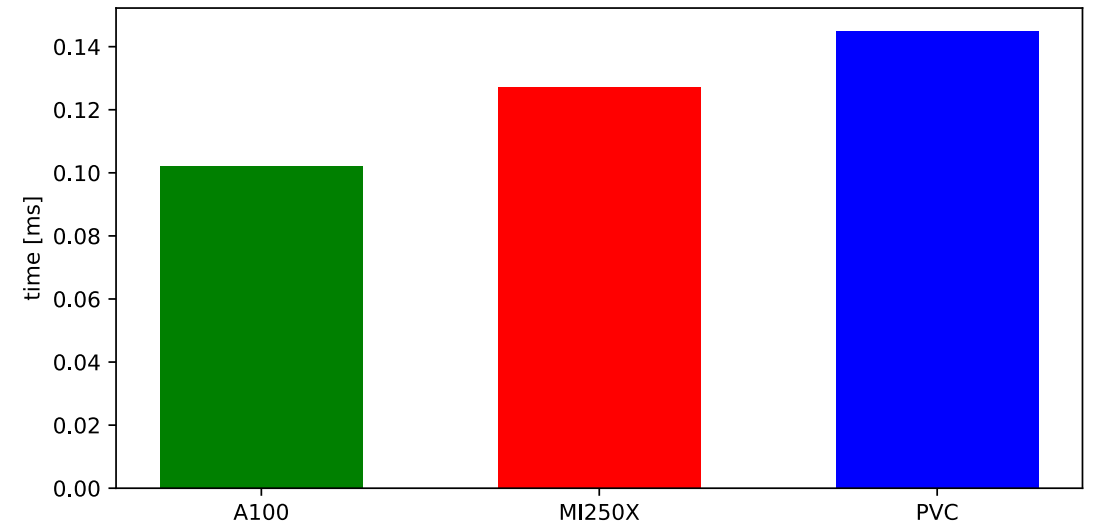
PLATFORM SUPPORT

HIP Backend

- HIP moves out of experimental in Kokkos 4.0.0
 - Kokkos::Experimental::HIP becomes Kokkos::HIP
 - Kokkos Kernels internal library clean-up
- More rocBLAS/rocSPARSE coverage
 - SpMV: single vector, multivector and block variants supported
 - SpGEMM and block SpGEMM
 - All Blas2/3 and most Blas1 supported
- Stream support using Kokkos execution space

SYCL Backend

- Still experimental, although more mature
- Almost all tests passing on Ponte Vecchio (still issue with SpGEMM)
- More TPL support of oneAPI MKL
- Nightly testing of SYCL, should promote to CI once stable and if testing capacity allows
- Integration with Trilinos and PETSc



Runs using ship_003 from SuiteSparse and Kokkos Kernels
native SpMV



NEW FEATURES

GENERAL LIBRARY UPDATES

- Library reorganized by components
 - Blas
 - Batched dense/sparse
 - Sparse
 - Graph
 - ODE (WIP)
- Added oneMKL TPL
- Kokkos Kernels version macros
 - CMake: KokkosKernels_VERSION
 - Header (KokkosKernels_config.h): KOKKOSKERNELS_VERSION

GENERAL LIBRARY UPDATES

- Google Benchmark TPL
 - Enable with: `KokkosKernels_ENABLE_TEST=ON` + `KokkosKernels_ENABLE_PerfTests=ON` + `KokkosKernels_ENABLE_Benchmarks=ON`
- Configuration output
 - `KokkosKernels::print_configuration(std::ostream&)`
 - Prints library version and TPL information
 - Feedback welcomed on what additional information should be printed!

BLAS

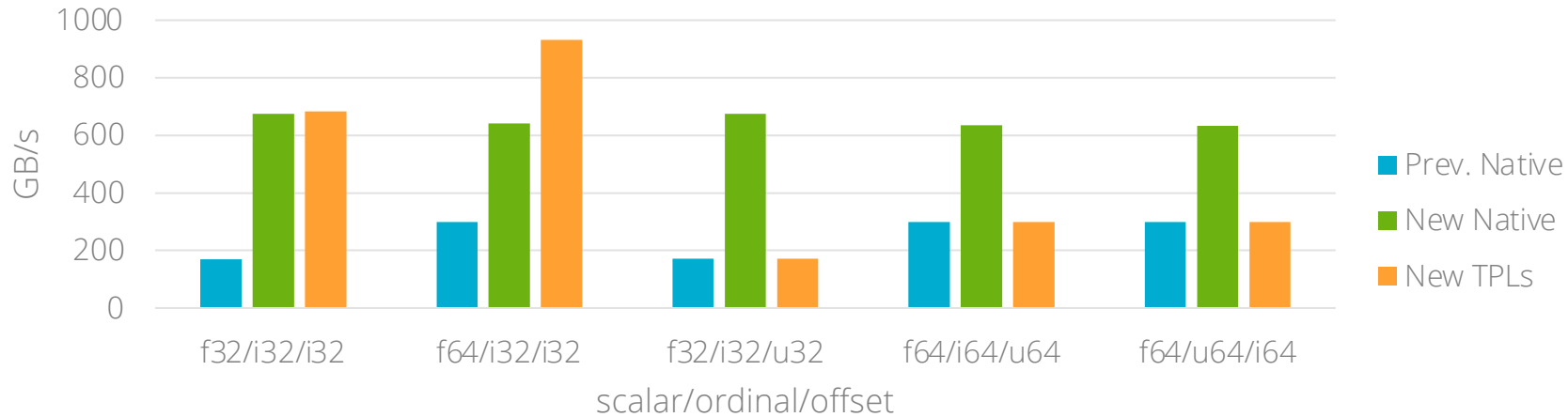
- Blas completeness
 - Blas1 complete
 - Blas2
 - General/Symmetric matrix needs SYMV to be complete
 - No packed/banded algo yet
 - Blas3
 - General/Symmetric: need SYMM, HEMM and rank k/2k updates
- All Blas algorithms support stream execution
 - KokkosBlas::myBlasKernel(space, ...);
- General maintenance of TPLs
 - Added support for newer versions of cuBLAS/rocBLAS
 - Working on oneMKL support for Intel GPUs

SPARSE

- Sparse format conversion
 - `coo2csr`, `csc2csr`
- Merged based SpMV for unbalanced rows in matrix
- SpGEMM
 - New “reuse” interface, saves graph of previous matrix
 - Improved TPL support (MKL, cuSPARSE) new rocSPARSE support
- Incomplete factorizations
 - New parILUt algorithm (iterative computation of L and U)
 - New MDF(0) algorithm (re-orders following Frobenius norm of discard factor on the fly)
 - Stream version of ILU(k) and SpTRSV

SPARSE

- Brs format support
 - Improved SpMV performance especially on AMD platform with TPLs
 - Results below: 1 vector, block size = 7, rocm 5.2.0

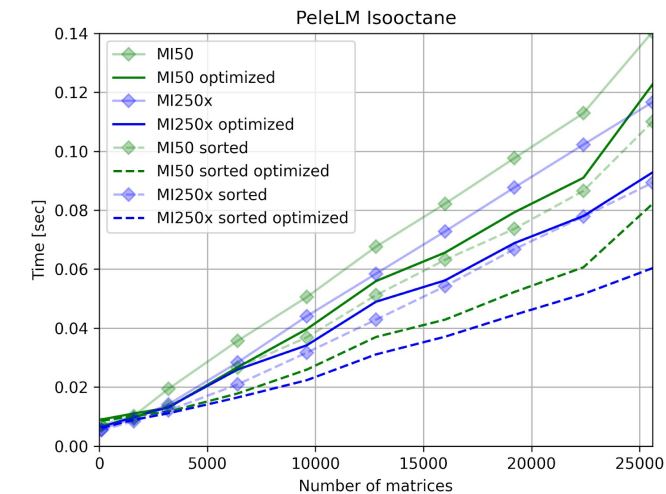
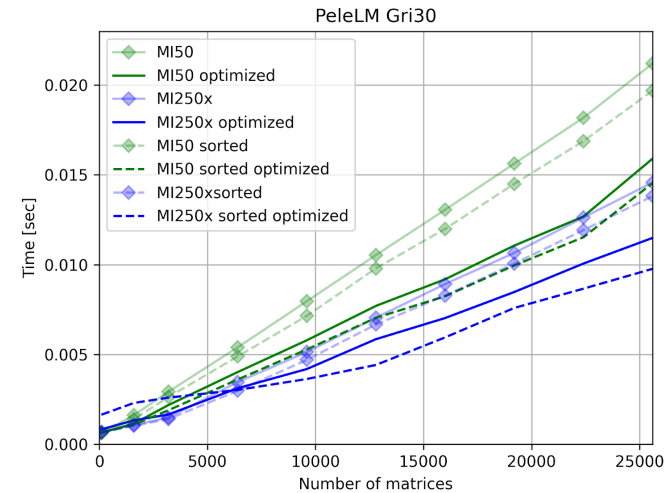


- CrsMatrix sort and merge
 - Needed for some TPL
 - Useful after SpGEMM and or MPI communication

BATCHED

Sparse Batched algorithms

- Algorithms implemented:
 - Linear algebra (SpMV)
 - Iterative solvers (CG, GMRES)
 - Preconditioner (Jacobi)
- Launch parameters tuned for architecture
 - NVIDIA V100
 - AMD MI50 / MI250

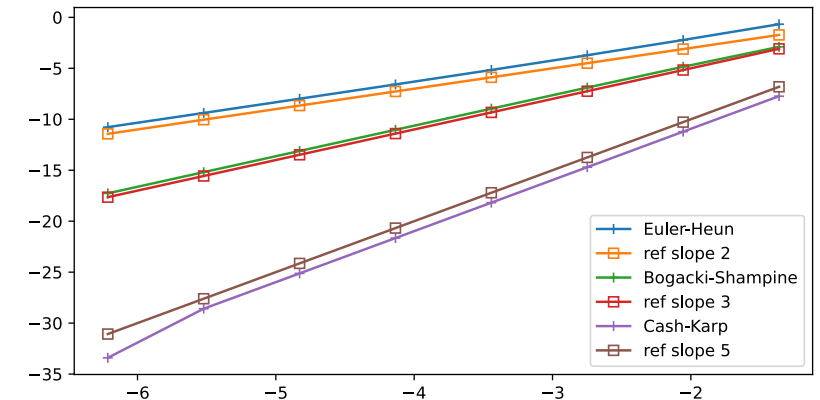


ODE

New component for time integration algorithms

- Explicit integrators
 - Runge Kutta (orders 1 to 5)
 - Various schemes for stability (Fehlberg 45, Cash-Karp, Dormand-Prince)
 - Time adaptive
- Implementation for GPU work within a RangePolicy

Num systems	10,000	100,000	1,000,000
CTS-1	3.17128	31.709	N/A
ATS-2	0.0303719	0.365714	2.33355



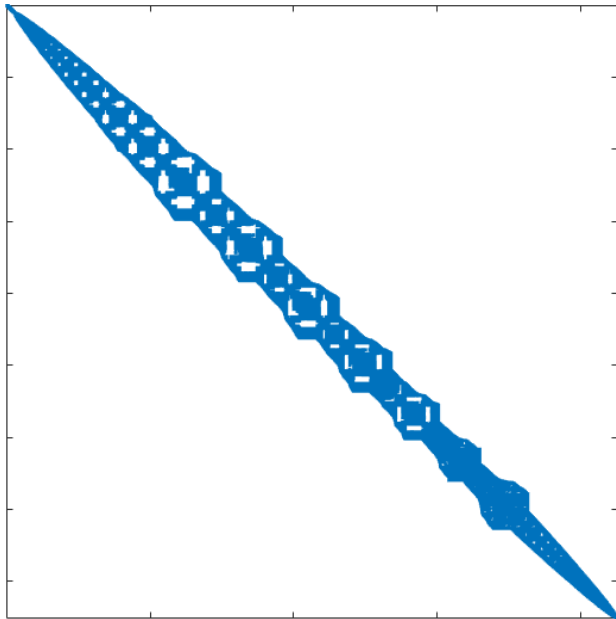


RELEASE 4.2.0

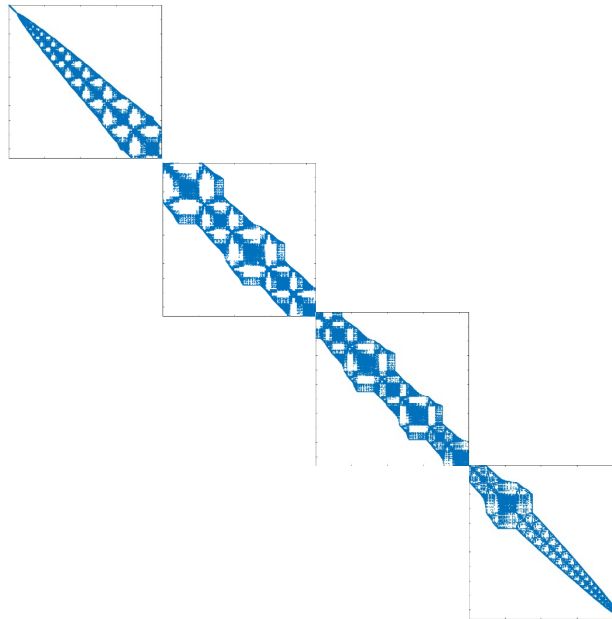
STREAM BASED SPARSE PRECONDITIONERS

- Stream Gauss-Seidel
- Stream ILU(k)/SpTRSV
 - Decompose the problem as we would with MPI
 - Use a stream per subdomain

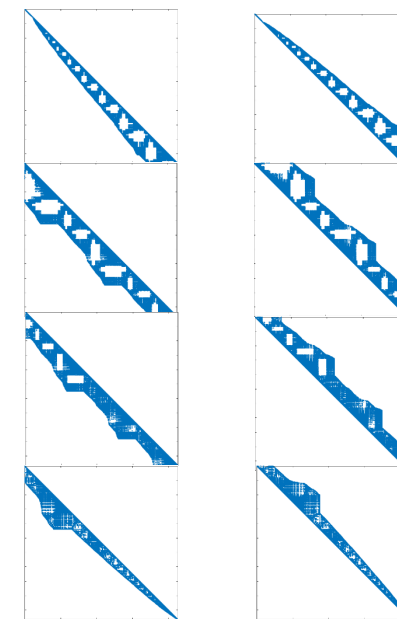
Local MPI rank matrix



Stream split matrices

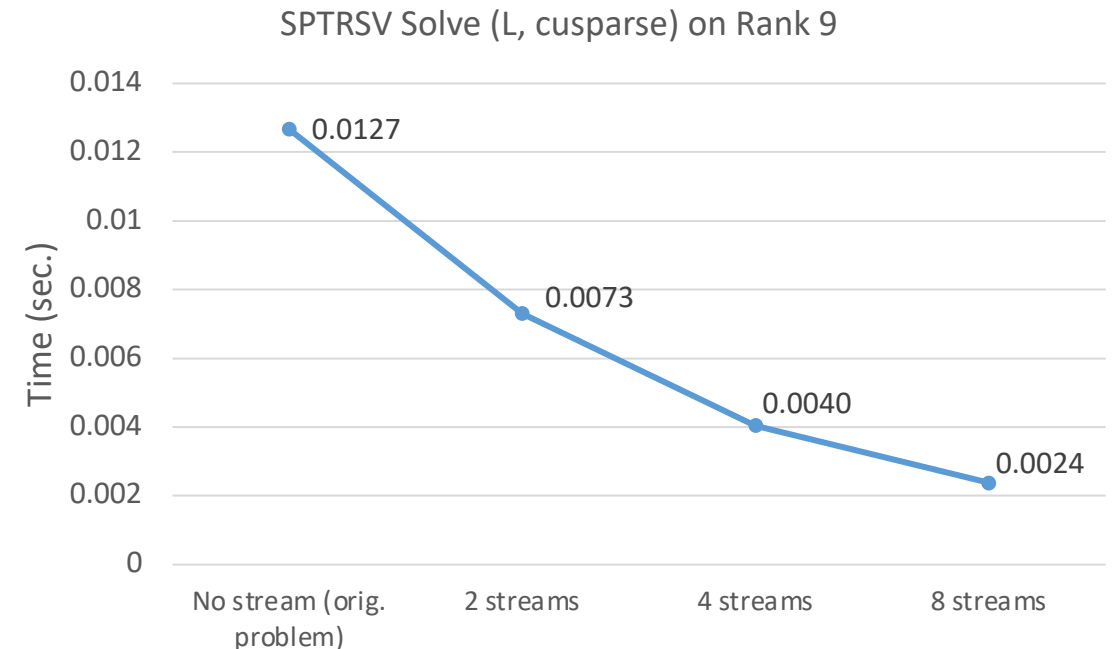
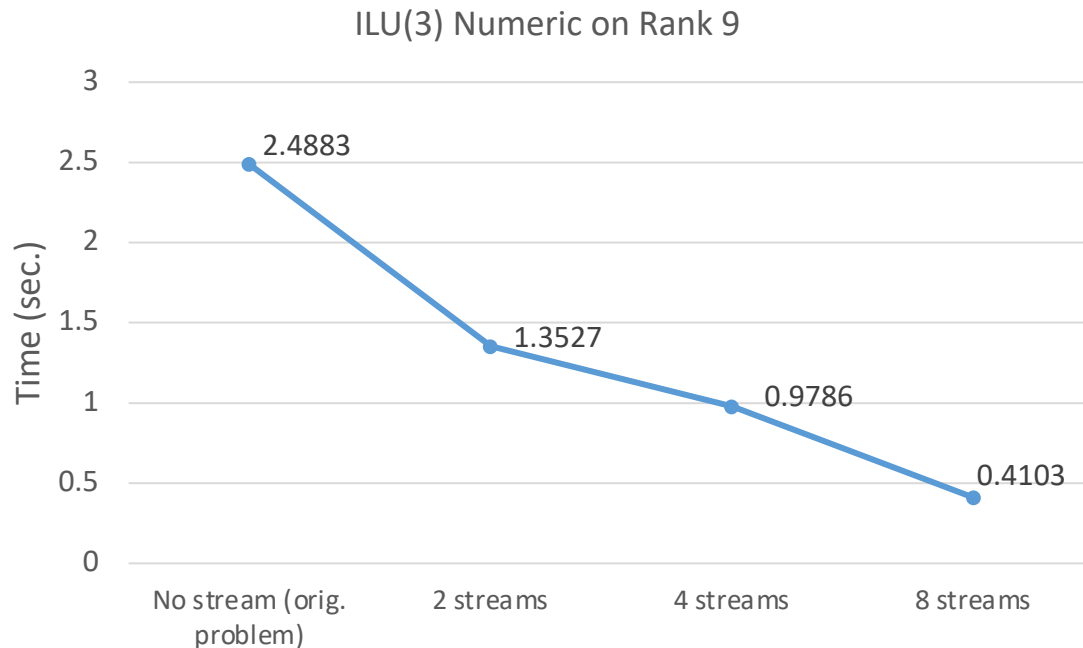


Stream split factors



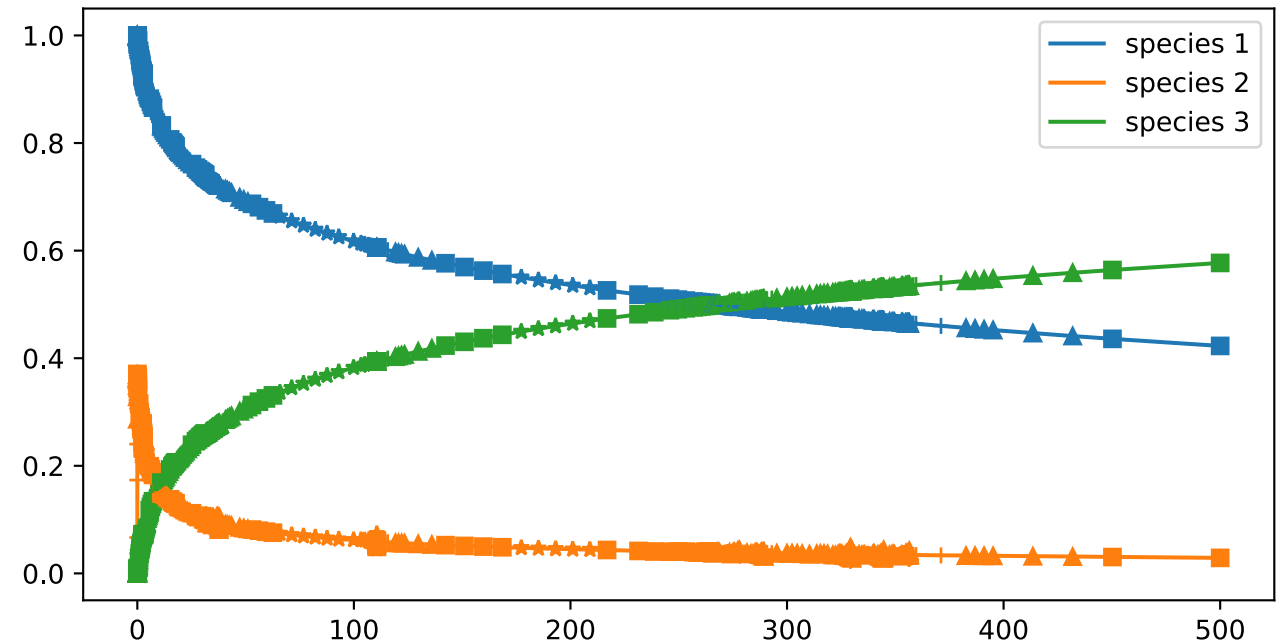
STREAM BASED ILU(K)

- Scaling study 1 to 8 streams
 - Good scaling overall
 - Some scalability loss in SpTRSV on 8 streams
 - Performance very dependent on CUDA version (results obtained with CUDA 11.8)
- Like MPI partitioning, balancing is important!



ODE

- Newton solver
 - Drive adaptation from convergence behavior
 - Cheaper secant variant option
- BDF, implicit time integration
 - Similar feature to CVODE
 - Time and order adaptive
 - Order 1 to 5
 - Initial time step estimation
- Best option for stiff problems see left





UPCOMING WORK

ALGORITHMIC DEVELOPMENT

- Block-ILU(k) variant
 - Fill based on block graph
 - Integration with Ifpack2
- LAPACK select algorithms implementation (LU, SVD, QR)
 - Add new library component
 - Include cuSOLVER, rocSOLVER, MKL and Magma TPLs
- Improve BDF features:
 - Numerical differentiation Jacobian
 - Backtracking line search
- Batched ODE solvers
 - Reduce branch divergence on GPU
 - Promote vectorization on CPU
 - Potentially complicated for BDF, easier for RK algorithms

LIBRARY IMPROVEMENTS

- SYCL backend: improve support and performance once Aurora comes online
- Improve integration with Trilinos and PETSc
- Establish automated performance testing
- Improve interface to enable auto-tuning

ACKNOWLEDGEMENTS

- Jonathan Hu and Tom Ransegnola for contributing multiple integrations of incomplete factorizations in Ifpack2 and more...
- Junchao Zhang for Kokkos Kernels/PETSc liaison, integration and contributing multiple TPL integrations and fixes
- Victor Brunini for interfacing with applications, providing design and performance feedback on new features
- Satish Balay and Sameer Shende for updating us on various incompatibilities and updates in Spack and xSDK
- Mark Adams for all the discussions on the batched linear solver interfaces and performance
- And to all the other contributors who help improved the library by providing feedback, documentation updates and bug fixes

We owe you a debt of gratitude, thank you for your continued support!



ANY
QUESTIONS?