

The objective of the Software for Linear Algebra Targeting Exascale (SLATE) project is to provide fundamental dense linear algebra capabilities to the US Department of Energy and to the high-performance computing (HPC) community at large. To this end, SLATE provides basic dense matrix operations (e.g., matrix multiplication, rank-k update, triangular solve), norms, linear systems solvers, least square solvers, and singular value and eigenvalue solvers.



The ultimate objective of SLATE is to replace the venerable Scalable Linear Algebra PACKage (ScaLAPACK) library, which has become the industry standard for dense linear algebra operations in distributed-memory environments. However, after two decades of operation, ScaLAPACK is past the end of its lifecycle and overdue for a replacement, as it can hardly be retrofitted to support hardware accelerators, which are an integral part of today's HPC hardware infrastructure. Primarily, SLATE aims to extract the full performance potential and maximum scalability from modern, many-node HPC machines with large numbers of cores and multiple hardware accelerators per node. For typical dense linear algebra workloads, this means getting close to the theoretical peak performance and scaling to the full size of the machine (i.e., thousands to tens of thousands of nodes). This is accomplished in a portable manner by relying on MPI, OpenMP, and C++ standards.

SLATE HIGHLIGHTS

Targets Modern Hardware

such as the upcoming exascale systems, where the number of nodes is large, and each node contains a heavyweight multi-core processor and a number of heavyweight hardware accelerators.

Guarantees Portability

by relying on standard computational components (e.g., vendor implementations of BLAS and LAPACK) and standard parallel programming technologies (e.g., MPI, OpenMP) or portable runtime systems (e.g., PaRSEC).

Provides Scalability

by employing proven techniques of dense linear algebra, such as the 2-D block cyclic data distribution, as well as modern parallel programming approaches, like dynamic scheduling and communication overlapping.

Facilitates Productivity

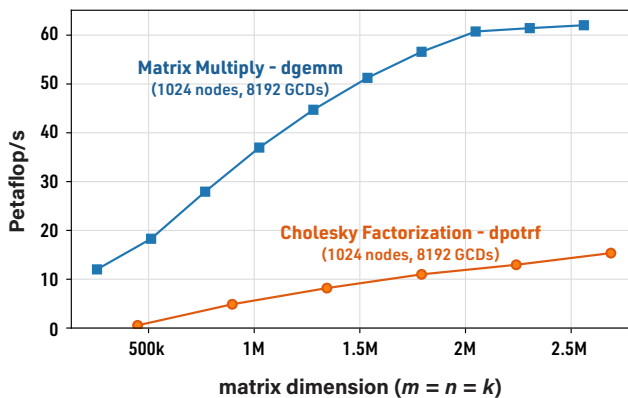
by relying on the intuitive single program, multiple data (SPMD) programming model and a set of simple abstractions to represent dense matrices and dense matrix operations.

Ensures Maintainability

by employing useful facilities of the C++ language, such as templates and overloading of functions and operators, and focused on minimizing code bloat by relying on compact representations.

PERFORMANCE

Performance of SLATE on 1024 Frontier nodes
4 AMD MI250X GPUs (8 GCDs) per node; 1 MPI process per GCD; GNU compiler



SLATEPAPERS and WORKING NOTES

<http://icl.utk.edu/slate/#papers>



Lindquist, Neil, Mark Gates, Piotr Luszczek, and Jack Dongarra. "Threshold Pivoting for Dense LU Factorization." In *2022 IEEE/ACM Workshop on Latest Advances in Scalable Algorithms for Large-Scale Heterogeneous Systems (ScalAH)*, pp. 34-42. IEEE, 2022.

Gates, Mark, Asim YarKhan, Dalal Sukkari, Kadir Akbudak, Sebastien Cayrols, Daniel Bielich, Ahmad Abdelfattah, Mohammed Al Farhan, and Jack Dongarra. "Portable and Efficient Dense Linear Algebra in the Beginning of the Exascale Era." In *2022 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*, pp. 36-46. IEEE, 2022.

SPONSORED BY



This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.

