



SLATE

REPLACING ScaLAPACK

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 will provide basic dense matrix operations (e.g., matrix multiplication, rank-k update, triangular solve), linear systems solvers, least square solvers, 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 to be accomplished in a portable manner by relying on standards like MPI and OpenMP.

SLATE HIGHLIGHTS

Targets Modern Hardware

such as the upcoming CORAL 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 (vendor implementations of BLAS and LAPACK), and standard parallel programming technologies (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, such as 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.

Assures Maintainability

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

SLATE WORKING NOTES

<http://www.icl.utk.edu/publications/series/swans>



SWAN003 Designing SLATE: Software for Linear Algebra Targeting Exascale

Kurzak, J., P. Wu, M. Gates, I. Yamazaki, P. Luszczek, G. Ragghianti, and J. Dongarra

SWAN004 C++ API for Batch BLAS

Abdelfattah, A., K. Arturov, C. Cecka, J. Dongarra, C. Freitag, M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Tomov, et al.

SWAN005 Parallel BLAS Performance Report,

Kurzak, J., M. Gates, A. YarKhan, I. Yamazaki, P. Wu, P. Luszczek, J. Finney, and J. Dongarra

FIND OUT MORE AT <http://icl.utk.edu/slate>

MAIN ECP WEBSITE
<https://exascaleproject.org>

SLATE REPOSITORY
<https://bitbucket.org/icl/slate>

BLAS++ REPOSITORY
<https://bitbucket.org/icl/blaspp>

LAPACK++ REPOSITORY
<https://bitbucket.org/icl/lapackpp>

RESEARCHGATE
<https://www.researchgate.net/project/ECP-SLATE>

SLATE MAILING LIST
slate-user@icl.utk.edu



Office of
Science



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.