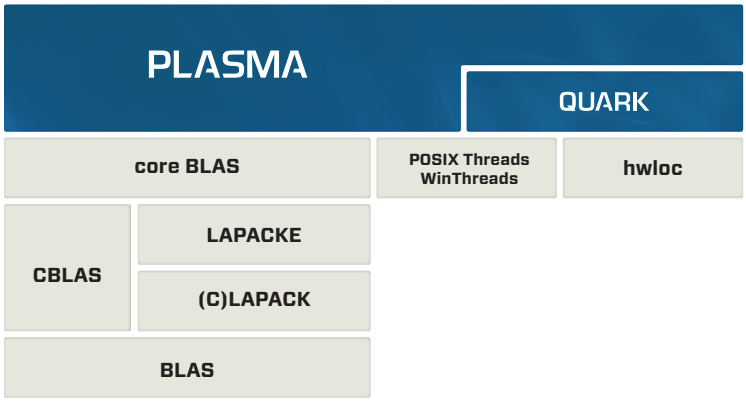
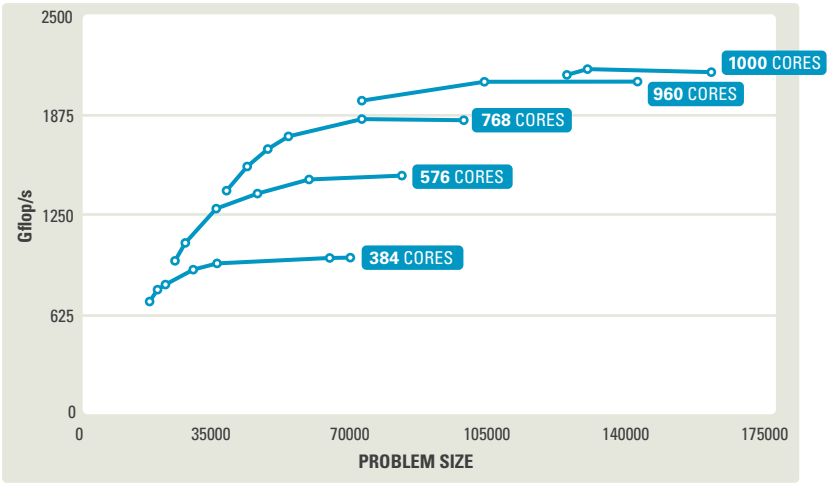
PLASMA

PARALLEL LINEAR ALGEBRA SOFTWARE FOR MULTICORE ARCHTECTURES

PLASMA (Parallel Linear Algebra Software for Multicore Archtectures) is a dense linear algebra package at the forefront of multicore computing. PLASMA is designed to deliver the highest possible performance from a system with multiple sockets of multicore processors. PLASMA achieves this objective by combining state-of-the-art solutions in parallel algorithms, scheduling and software engineering. Currently PLASMA offers a collection of routines for solving linear systems of equations, least square problems, eigenvalue problems and singular value problems.



Solving Symmetric Positive Definite System (DPOSV)
SGI Altix UV, 2.0 GHz Intel Nehalem EX System



TILE MATRIX LAYOUT

PLASMA lays out matrices in square tiles of relatively small size, such that each tile occupies a continuous memory region. Tiles are loaded to the cache memory efficiently with little risk of eviction while being processed. The use of tile layout minimizes conflict cache misses, TLB misses, and false sharing, and maximizes the potential for prefetching. PLASMA contains parallel and cache efficient routines for converting between the conventional LAPACK layout and the tile layout.

TILE ALGORITHMS

PLASMA introduces new algorithms redesigned to work on tiles, which maximize data reuse in the cache levels of multicore systems. Tiles are loaded to the cache and processed completely before being evicted back to the main memory. Operations on small tiles create fine grained parallelism providing enough work to keep a large number of cores busy.

DYNAMIC SCHEDULING

PLASMA relies on runtime scheduling of parallel tasks. Runtime scheduling is based on the idea of assigning work to cores based on the availability of data for processing at any given point in time and thus is also referred to as data-driven scheduling. The concept is related closely to the idea of expressing computation through a task graph,often referred to as the DAG (Directed Acyclic Graph), and the flexibility of exploring the DAG at runtime. This is in direct opposition to the fork-and-join scheduling, where artificial synchronization points expose serial sections of the code and multiple cores are idle while sequential processing takes place.

FIND OUT MORE AT http://icl.eecs.utk.edu/plasma/

INNOVATIVE COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE

A COLLABORATION OF
THE UNIVERSITY of TENNESSEE · Berkeley UNIVERSITY OF CALIFORNIA · University of Colorado Denver

WITH SUPPORT FROM
MathWorks · Microsoft · NVIDIA

SPONSORED BY
NSF

CITR
CENTER FOR INFORMATION TECHNOLOGY RESEARCH

PLASMA

FUNCTIONALITY	COVERAGE
Linear Systems of Equations	Cholesky, LDLT, LU with partial pivoting
Matrix Inversion	Cholesky, LU with partial pivoting
Least Squares	QR and LQ
Mixed Precision Iterative Refinement	linear systems using Cholesky or LU, least squares using QR or LQ
Symmetric Eigenvalue Problem	eigenvalues only
Singular Value Problem	singular values only
Level 3 Tile BLAS	GEMM, HEMM, HER2K, HERK, SYMM, SYR2K, SYRK, TRMM, TRSM
In-Place Layout Translation	CM, RM, CCRB, CRRB, RCRB, RRRB

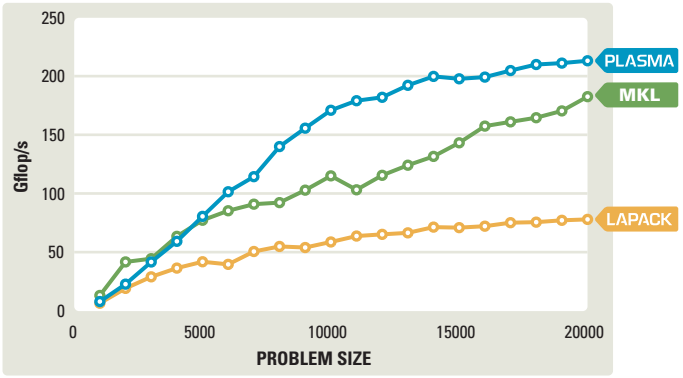
FEATURES

Covering four precisions:
double real, double complex, single real, single complex (D, Z, S, C)

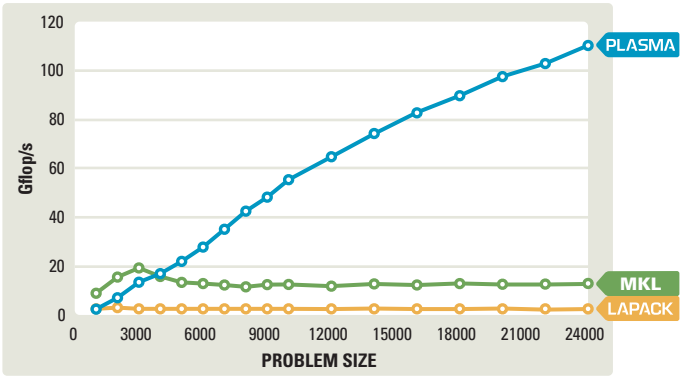
Providing LAPACK-compatible interface for matrices in F77 column-major layout

Supporting:
Linux, MS Windows, Mac OS, AIX

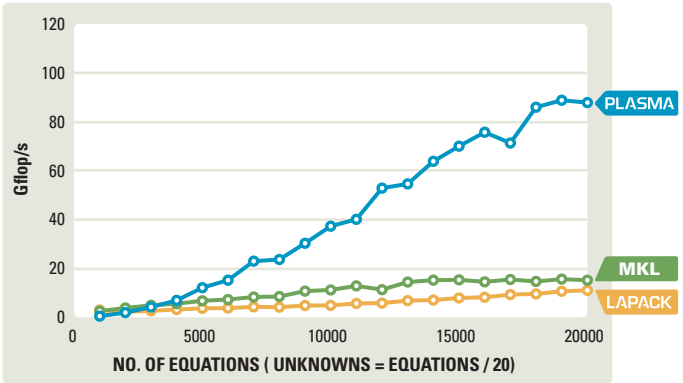
Solving Linear System (DGESV)
48-core, 2.1 GHz AMD Magny-Cours System



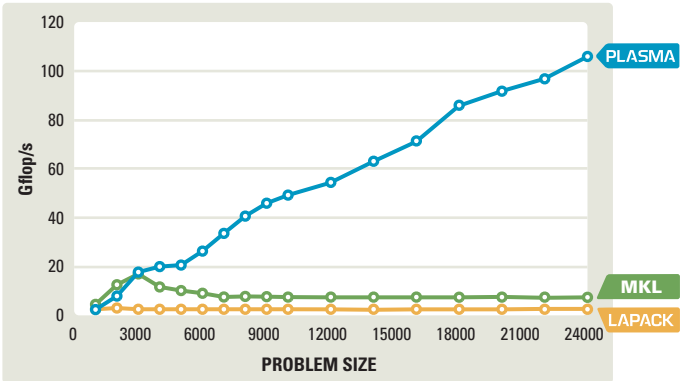
Solving Symmetric EVP (DSYEV)
48-core, 2.1 GHz AMD Magny-Cours System



Solving Least Squares Problem (DGELS)
48-core, 2.1 GHz AMD Magny-Cours System



Solving Singular Value Problem (DGESVD)
48-core, 2.1 GHz AMD Magny-Cours System



FIND OUT MORE AT http://icl.eecs.utk.edu/plasma/

INNOVATIVE COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE

A COLLABORATION OF
THE UNIVERSITY of TENNESSEE · Berkeley · University of Colorado Denver

WITH SUPPORT FROM
MathWorks · Microsoft · NVIDIA

SPONSORED BY
CITR CENTER FOR INFORMATION TECHNOLOGY RESEARCH