PARALLEL LINEAR ALGEBRA SOFTWARE FOR MULTICORE ARCHITECTURES

PLASMA

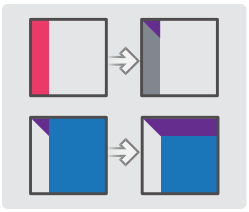
INNOVATIVE COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE

THE PARALLEL LINEAR ALGEBRA SOFTWARE FOR MULTICORE ARCHITECTURES (PLASMA) PROJECT aims to address the critical and highly disruptive situation that is facing the Linear Algebra and High Performance Computing community due to the introduction of multicore architectures. PLASMA's ultimate goal is to create software frameworks that enable programmers to simplify the process of developing applications that can achieve both high performance and portability across a range of new architectures. PLASMA uses a programming model that allows asynchronous, out-of-order scheduling of operations in order to achieve a scalable yet highly efficient software framework for Computational Linear Algebra applications.

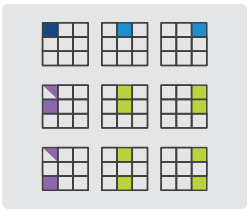
TILE ALGORITHMS

Unlike LAPACK, which uses block algorithms, PLASMA relies on tile algorithms to enable the use of fine grained parallelism.

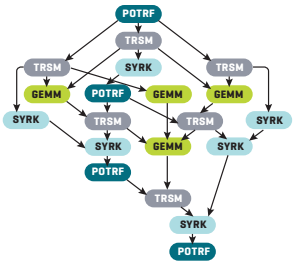
LAPACK
Block algorithms



PLASMA
Tile algorithms



Tile algorithms of Linear Algebra operations can be represented as Directed Acyclic Graphs (DAG) where nodes represent the tasks in which the operation can be decomposed and the edges represent the dependencies among them. As long as the task execution order does not violate the dependencies, the result will be correct.



Example of a DAG for a Cholesky Factorization

PLASMA 2.1.0

- Solution of Linear Equations
- Linear Least Squares Problems
- Multiple Precision Support
- Mixed-Precision Iterative Solver
- Static Scheduling
- LAPACK Interface / Native Interface
- LAPACK-Compliant Error Handling
- LAPACK-Derived Testing Suite
- Thread Safety
- Windows, Linux, AIX, Mac OS
- PLASMA Users' Guide

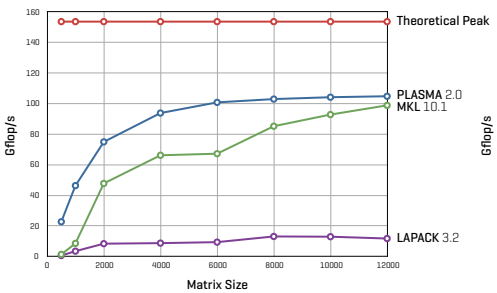
CURRENT RESEARCH

- Singular Value Decomposition
- Symmetric and Non Symmetric Eigenvalue Problems
- Dynamic Scheduling
- Communication Avoiding Algorithms
- Autotuning
- Distributed Memory Machines
- Hardware Accelerators

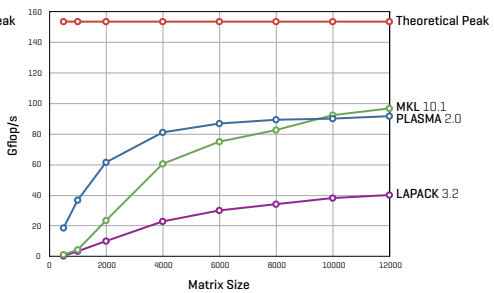
PERFORMANCE RESULTS DOUBLE PRECISION

CPU Intel Xeon 2.4 GHz Quad-socket Quad cores (16 cores total)

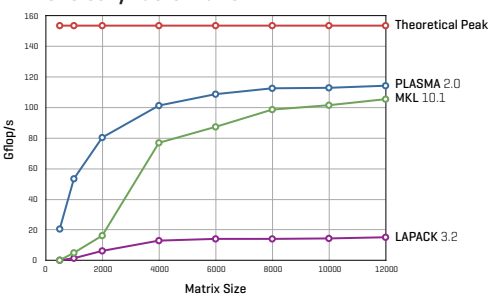
QR Factorization



LU Factorization



Cholesky Factorization



DOWNLOAD THE LIBRARY AT http://icl.eecs.utk.edu/plasma/

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

WITH SUPPORT FROM
The MathWorks · Microsoft

SPONSORED BY
CITR CENTER FOR INFORMATION TECHNOLOGY RESEARCH