MAGMA

MATRIX ALGEBRA ON GPU AND MULTICORE ARCHITECTURES

The MAGMA project, led by the linear algebra research groups at University of Tennessee, UC Berkeley, and UC Denver, aims to develop a linear algebra library similar to LAPACK but for heterogeneous/hybrid architectures, starting with current "Multicore+GPU" systems. This transition cannot be done automatically, as in many cases new algorithms that significantly differ from algorithms for conventional architectures will be needed. Preliminary studies on a new class of "heterogeneity-aware" algorithms of "reduced communication" and "high-parallelism" confirm that this is the case.

HARDWARE TO SOFTWARE TRENDS

Hardware

- INCREASE IN PARALLELISM
- INCREASE IN COMMUNICATION COST (vs COMPUTATION)

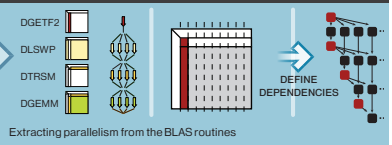
Hybrid / Heterogeneous Designs

Multicore + GPUs

NEW ALGORITHMS NEEDED

Software

- EXTRACT PARALLELISM
- REDUCE COMMUNICATION

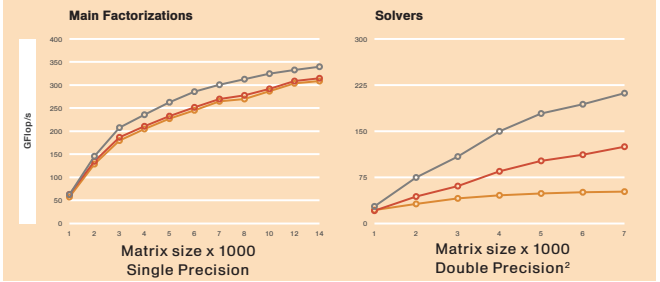


Extracting parallelism from the BLAS routines

- HETEROGENEITY-AWARE ALGORITHMS
- INNOVATIVE DATA STRUCTURES
- PRECONDITIONING TO REDUCE PIVOTING
- MIXED PRECISION ALGORITHMS

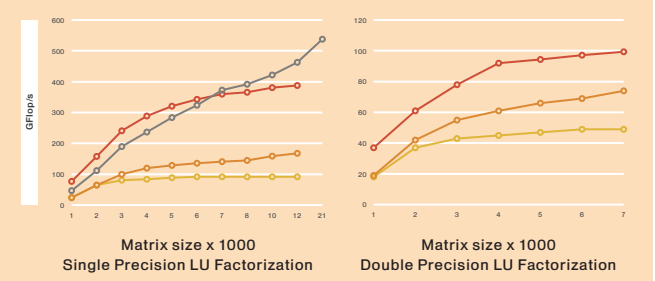
PERFORMANCE RESULTS[1]

Single Core + Single GPU



- QR
- Cholesky
- PP LU
- Iterative Mixed Precison (GPU Iterations)
- Iterative Mixed Precision (CPU Iterations)
- Direct Solver

Multicores / Multi-GPUs



- 2 Cores + 2 GPUs [Partial Pivoting (PP) LU]
- 8 Cores + 1 GPU [Preconditioned Limited Pivoting (LP) LU(NB)[3]]
- 16 Cores / Tigerton [Pairwise Pivoting (PwP) LU from PLASMA]
- 8 Cores / Harpertown [PwP LU from PLASMA]

HARDWARE USED

GPU: GeForce GTX 280 (240 Cores @ 1.30 GHz) Host: Intel Xeon (2 x 4 Cores @ 2.33 GHz) Tigerton: Intel Xeon (4 x 4 Cores @ 2.4 GHz) Harpertown: Intel Xeon (2 x 4 Cores @ 2.33 GHz)

[1] The new techniques often gain in speed for the price of reduced accuracy. Understanding this trade-off of speed vs accuracy can lead to very efficient algorithms.

[2] Mixed precision solvers often achieve 4 x speedup compared to DP solvers but the speed depends on the conditioning of the matrix. In these performance results, we considered three steps of iterative refinement (on symmetric and positive definite matrices using Cholesky).

[3] Limited amount of pivoting (within the block size NB or more) is justified by a specially designed unitary transformation: experiments with random matrices show that LP LU(NB+64) for example is comparable in accuracy to PP LU, and LP LU(NB) loses only from 1 to 2 digits of accuracy to gain up to 30% in speed compared to PP LU.

SPONSORED BY The MathWorks Microsoft NVIDIA NSF

University of California Berkeley
University of Coimbra FCTUC
University of Colorado Denver Downtown Campus
The Parallel Computing Laboratory

The University of Tennessee | Innovative Computing Laboratory | Center for Information Technology Research