MAGMA: Matrix Algebra on GPU and Multicore Architectures

Presented by

Scott Wells

Assistant Director
Innovative Computing Laboratory (ICL)
College of Engineering
University of Tennessee, Knoxville

Overview

- **MAGMA:** a new generation of linear algebra (LA) libraries to achieve the fastest possible time to an accurate solution on hybrid/heterogeneous architectures, starting with current multicore + multiGPU systems
  Homepage: http://icl.cs.utk.edu/magma/

- **MAGMA & LAPACK**
  - MAGMA – based on LAPACK and extended for hybrid systems (multicore + multiGPU systems)
  - MAGMA – designed to be similar to LAPACK in functionality, data storage, and interface, to allow scientists to effortlessly port any LAPACK-relying software components to take advantage of new architectures
  - MAGMA – to leverage years of experience in developing open source LA software packages and systems like LAPACK, ScaLAPACK, BLAS, and ATLAS, as well as the newest LA developments (e.g., communication avoiding algorithms) and experiences on homogeneous multicores (e.g., PLASMA)

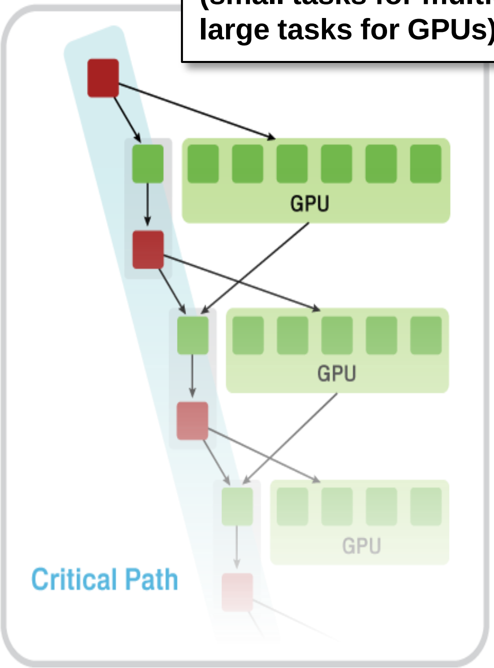
- **Support**
  - NSF, Microsoft, NVIDIA (CUDA Center of Excellence at UTK on the development of Linear Algebra Libraries for CUDA-based Hybrid Architectures)

- **MAGMA developers**
  - University of Tennessee, Knoxville; University of California, Berkeley; University of Colorado, Denver

Methodology overview

- MAGMA uses hybridization methodology based on
  - Representing linear algebra algorithms as collections of tasks and data dependencies among them
  - Properly scheduling tasks' execution over multicore and GPU hardware components
- Successfully applied to fundamental linear algebra algorithms
  - One- and two-sided factorizations and solvers
  - Iterative linear and eigensolvers
- Faster, cheaper, better
  - High level
  - Leveraging prior developments
  - Exceeding in performance homogeneous solutions

Hybrid CPU+GPU algorithms
(small tasks for multicores and large tasks for GPUs)

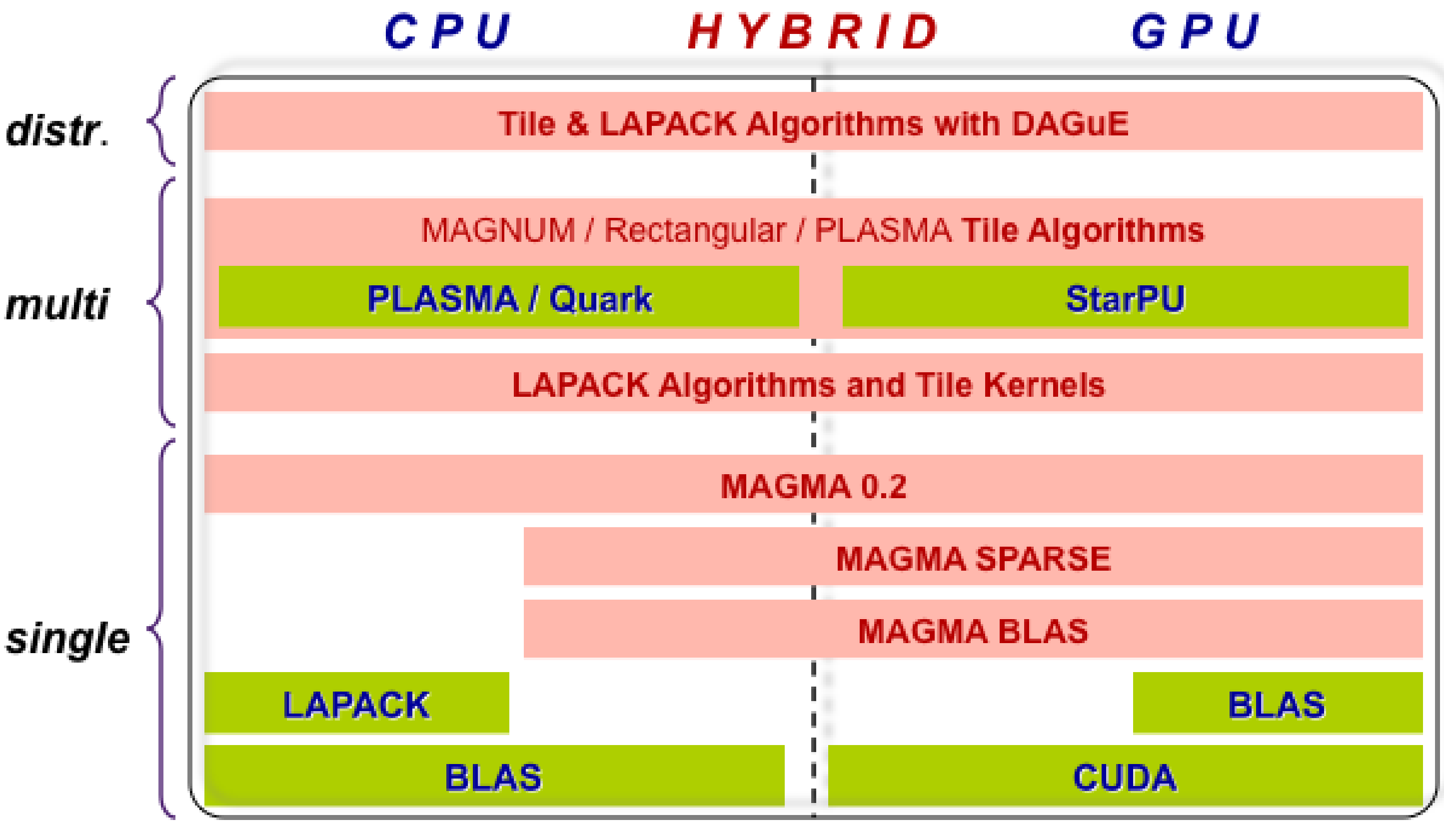


Managed by UT-Battelle
for the U.S. Department of Energy

MAGMA status

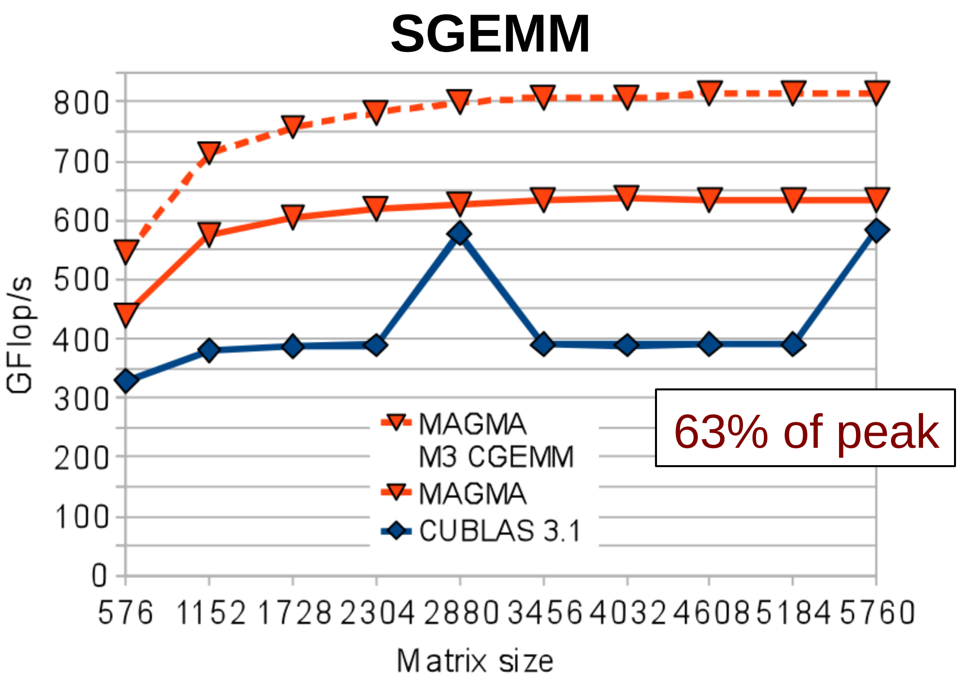
- One-sided factorizations
  - LU, QR, LQ, and Cholesky (S, C, D, Z)
- Linear solvers
  - In working precision, based on LU, QR, LQ, and Cholesky
  - Mixed-precision iterative refinement
- CPU and GPU interfaces
- MAGMA BLAS
  - Routines critical for MAGMA (GEMM, SYRK, TRSM, GEMV, SYMV, etc.)
  - BLAS for Fermi
- Two-sided factorizations
  - Reduction to upper Hessenberg form for the general eigenvalue problem
  - Tridiagonal for the symmetric eigenvalue problem
  - Bidiagonal reduction for SVD
- Divide & Conquer for the symmetric eigenvalue problem
- Algorithms for multiGPU and multicore use
- GMRES and PCG

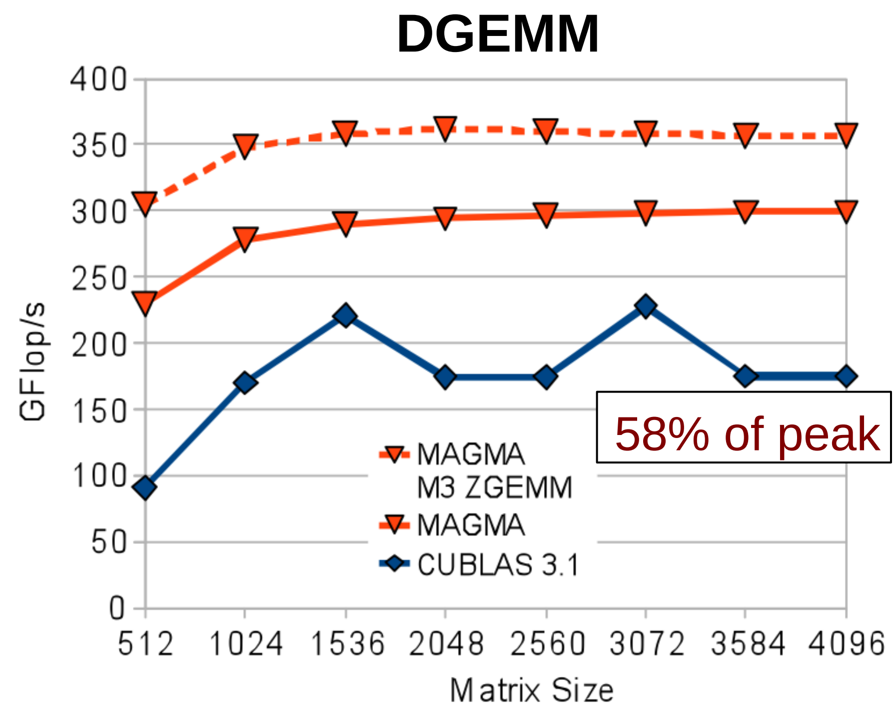
MAGMA software stack



Linux, Windows, Mac OS X | C/C++, Fortran | Matlab, Python

Results – BLAS



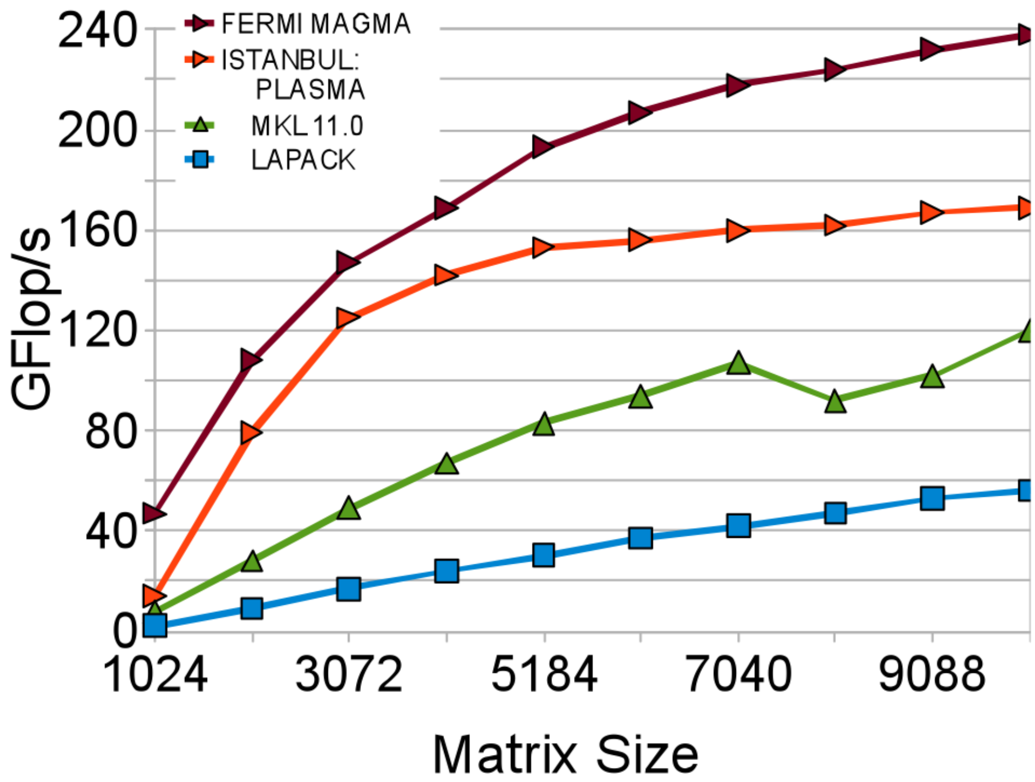


Tesla C2050 (Fermi): 448 CUDA cores @ 1.15 GHz; theoretical SP peak, 1.03 Tflop/s; DP peak, 515 GFlop/s)

- **TRSM and other Level 3 BLAS based on GEMM**
- **Use other hardware (e.g., ATI) through OpenCL**
  - **Based on auto-tuning various parameterized kernels**
- **"Auto-tuning" has become more important**
  - **e.g., for BLAS, higher-level hybrid algorithms, OpenCL port**

Results – one-sided factorizations

LU factorization in double precision



FERMI	Tesla C2050: 448 CUDA cores @ 1.15 GHz SP/DP peak is 1030; 515 Gflop/s (system cost ~$3,000)
ISTANBUL	AMD 8 socket 6 core (48 cores) @ 2.8 GHz SP/DP peak is 1075; 538 Gflop/s system cost ~$30,000)

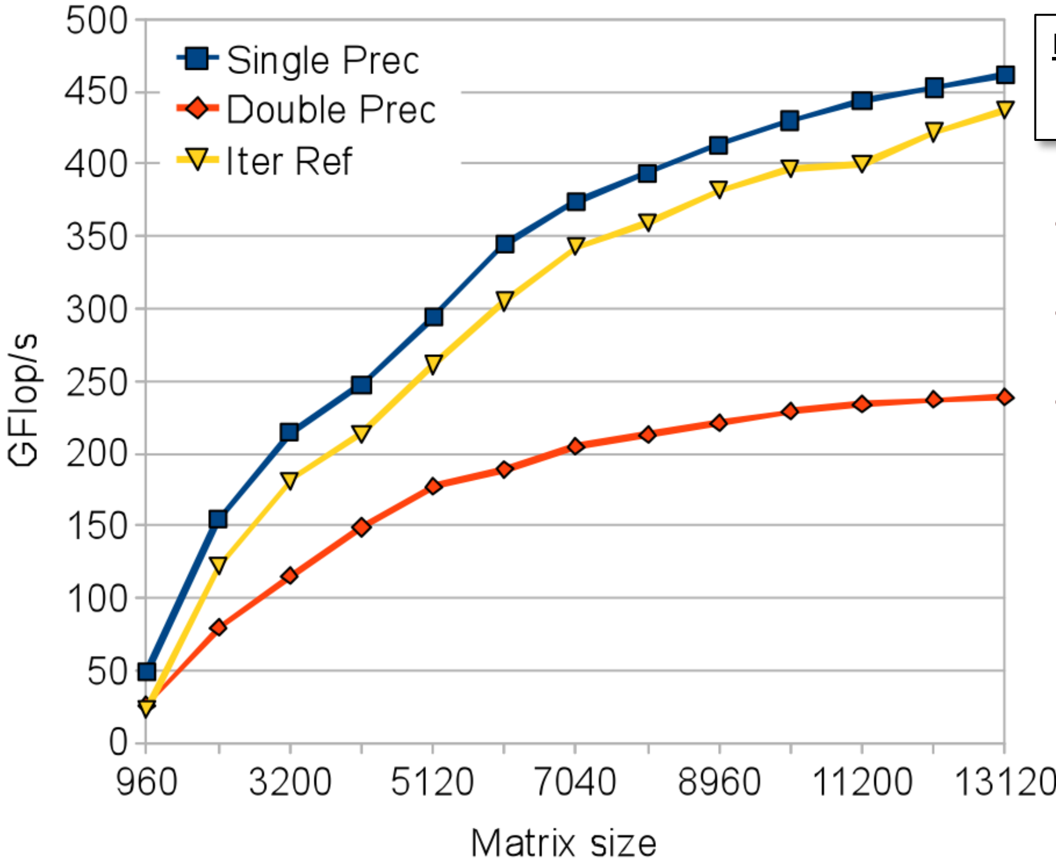
- **Similar results for Cholesky & QR**
- **60% faster than the commercially available CULA library for GPUs**
- **Fast solvers (several innovations)**
  - **In working precision, and**
  - **Mixed-precision iter. refinement based on the one-sided factor**

 Managed by UT-Battelle
for the U.S. Department of Energy

OAK RIDGE National Laboratory

Results – linear solvers

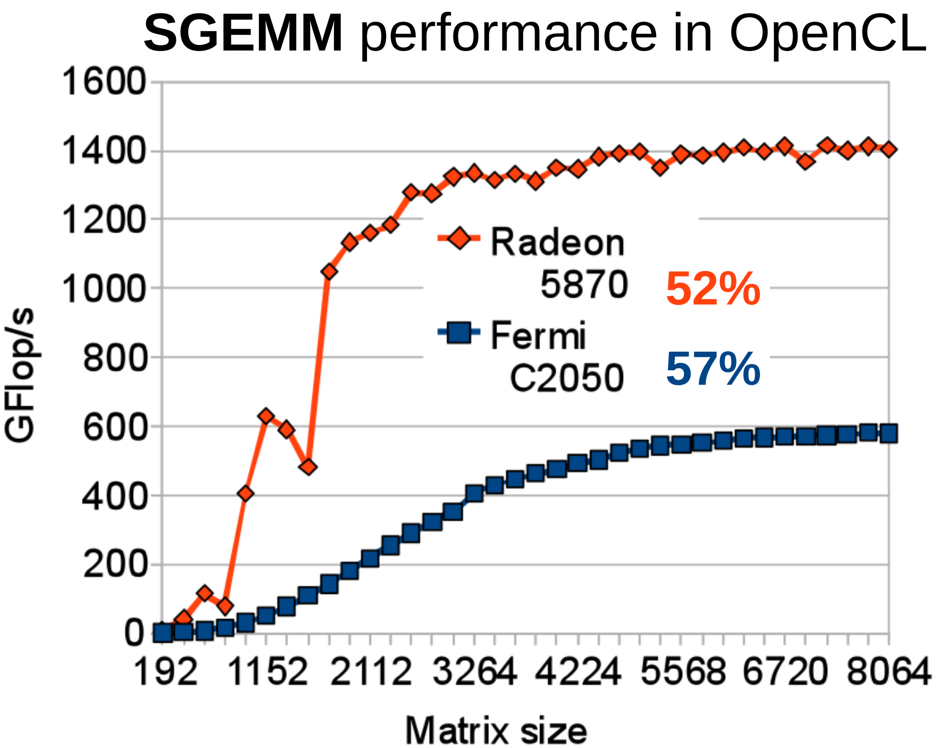
MAGMA LU-based solvers on Fermi (C2050)

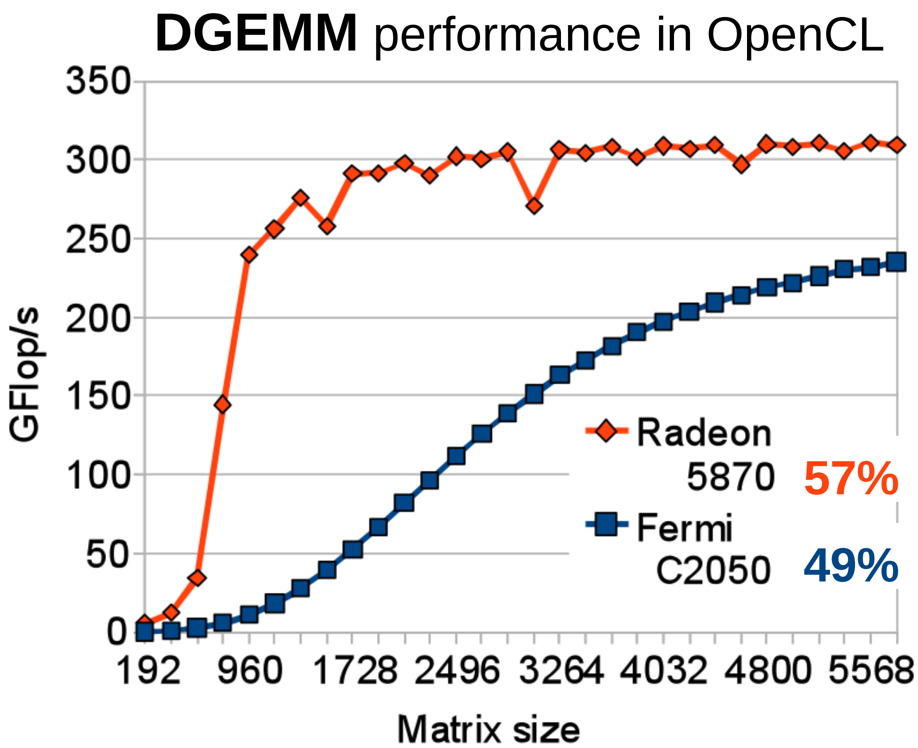


FERMI Tesla C2050: 448 CUDA cores @ 1.15 GHz
SP/DP peak is 1030; 515 Gflop/s

- **Similar results for Cholesky & QR**
- **60% faster than CULA**
- **Fast solvers (several innovations)**
  - **In working precision, and**
  - **Mixed-precision iter. refinement based on the one-sided factor**

Results – portability across platforms through OpenCL

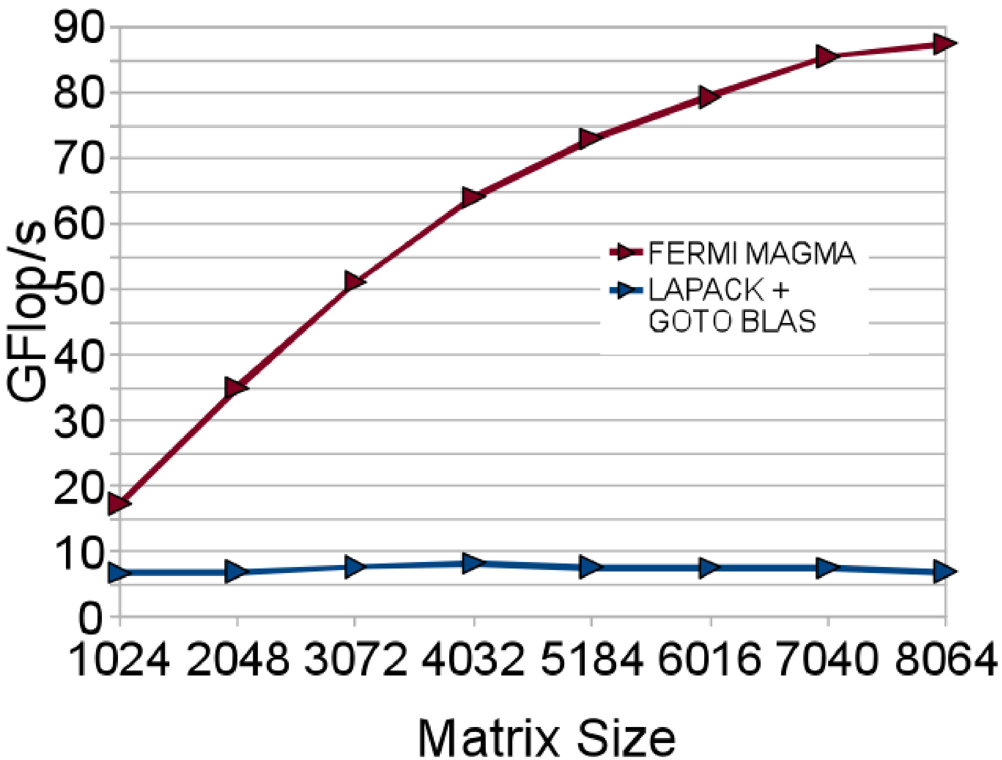




- **Performance portability of OpenCL implementations**
  - **Trough auto-tuning**
  - **Collecting best kernel versions**
  - **Generating multiple kernel versions to explore the kernel parameter space**
  - **Find best performing kernel versions on particular architecture using empirical-based search enhanced with heuristic models**

Results – two-sided factorizations

Hessenberg factorization in double precision (for the general eigenvalue problem)



FERMI	Tesla C2050: 448 CUDA cores @ 1.15 GHz SP/DP peak is 1030; 515 Gflop/s (system cost ~$3,000)
ISTANBUL	AMD 8 socket 6 core (48 cores) @ 2.8 GHz SP/DP peak is 1075; 538 Gflop/s system cost ~$30,000)

- **Similar accelerations for the bidiagonal factorization (for SVD) and tridiagonal factorization (for the symmetric eigenvalue problem)**
- **Similar acceleration (exceeding 10x) compared to other top-of-the-line multicore systems (including Nehalem-based) and libraries (including MKL, ACML)**

Contact

Scott Wells

swells@eecs.utk.edu

MAGMA team

http://icl.cs.utk.edu/magma

PLASMA team

http://icl.cs.utk.edu/plasma

Collaborating partners

University of Tennessee, Knoxville
University of California, Berkeley
University of Colorado, Denver
University of Coimbra, Portugal
INRIA, France (StarPU team)

Microsoft

NVIDIA

The MathWorks

NSF

U.S. DEPARTMENT OF ENERGY