MAGMA - Current and future work

- Algorithms for Multicore + GPU
  - Where performance is % of Multicore peak + GPU peak
- Release the two-sided factorizations
- Complete eigen-solvers
- Communication-optimal algorithms
- User-defined accuracy
  - trade-off accuracy for speed; mixed-precision solvers
- CUDA BLAS kernels
- Portability – demonstrate an easy OpenCL port
- Sparse linear algebra kernels
  - SpMV for structured (e.g. stencils) and unstructured matrices; iterative linear/eigen-solvers
One and two-sided Multicore+GPU Factorizations

These will be included in up-coming MAGMA releases

**Two-sided factorizations can not be efficiently accelerated on** homogeneous x86-based **multicores** (above) because of memory-bound operations
- we developed **hybrid algorithms that overcome those bottlenecks** (16x speedup!)
MAGMA BLAS

- Accelerate a subset of BLAS that is crucial to the performance of MAGMA routines
  - GEMM on rectangular matrices and sizes not divisible by certain [currently best performing] block sizes
  - approach is based on auto-tuning
  - important for all routines
  - Work with triangular matrices, e.g. TRSM
    - important for many routines
    - sometimes can be avoided
    - crucial for example in mixed-precision iterative solvers while iterating on the GPU
  - GEMV
    - used in Hessenberg and mixed-precision iterative solvers
GEMM Acceleration

- GEMM on rectangular matrices
  - various kernels needed for the one and two-sided factorizations, e.g.
    \[ \text{magma}_\text{dgemm}('n', 'n', n-k, n-k-32, 32, \ldots) \]

- Remove performance oscillations
MAGMA TRSM and GEMV

**STRSM**

Strsm Performance
Lx=B, ['L', 'L', 'N', 'N']

- GPU: GeForce GTX 280
- CPU: 2x Intel Xeon E5410 Quad Core (2.33GHz)
- NRHS [columns of B]: 16
- Thread Block: 32 x 1
- CPU threads: 8

**DGEMV**

Matrix size x 1,000

- MAGMA
- CUBLAS
- Multicore

Graphs comparing STRSM and DGEMV performance across different matrix sizes.