MAGMA (Matrix Algebra on GPU and Multicore Architectures) is a collection of next generation linear algebra libraries for heterogeneous architectures. MAGMA is designed and implemented by the team that developed LAPACK and ScaLAPACK, incorporating the latest developments in hybrid synchronization- and communication-avoiding algorithms, as well as dynamic runtime systems. Interfaces for the current LAPACK and BLAS standards are supported to allow computational scientists to seamlessly port any linear algebra reliant software components to heterogeneous architectures. MAGMA allows applications to fully exploit the power of current heterogeneous systems of multi/many-core CPUs and multi-GPUs to deliver the fastest possible time to accurate solution within given energy constraints.

HYBRID ALGORITHMS
MAGMA uses a hybridization methodology where algorithms of interest are split into tasks of varying granularity and their execution scheduled over the available hardware components. Scheduling can be static or dynamic. In either case, small non-parallelizable tasks, often on the critical path, are scheduled on the CPU, and larger more parallelizable ones, often Level 3 BLAS, are scheduled on the GPU.

PERFORMANCE & ENERGY EFFICIENCY

MAGMA LU factorization in double precision arithmetic

<table>
<thead>
<tr>
<th>CPU</th>
<th>NVIDIA K40 GPU</th>
<th>NVIDIA Pascal GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>2x10 cores</td>
<td>15 MP x 192 @ 0.88 GHz</td>
<td>56 MP x 64 @ 1.19 GHz</td>
</tr>
</tbody>
</table>

Features and Support

- MAGMA 2.2 for CUDA
- cIMAGMA 1.4 for OpenCL
- MAGMA MIC 1.4 for Intel Xeon Phi

Industry Collaboration

NVIDIA’s GPU Center of Excellence Program recognizes universities expanding the frontier of massively parallel computing using CUDA.

Intel Parallel Computing Center

The objective of the Innovative Computing Laboratory’s IPCC is the development and optimization of numerical linear algebra libraries and technologies for applications, while tackling current challenges in heterogeneous Intel® Xeon Phi™ coprocessor-based High Performance Computing.

In Collaboration With

INNOVATIVE COMPUTING LABORATORY

With Support From

SPONSORED BY

U.S. Department of Defense

National Science Foundation

SILAS Award

Find Out More At http://icl.utk.edu/magma
MAGMA BATCHED

BATCHED FACTORIZATION OF A SET OF SMALL MATRICES IN PARALLEL

Numerous applications require factorization of many small matrices

- Deep learning
- Structural mechanics
- Astrophysics
- Sparse direct solvers
- High-order FEM simulations

Routines

- LU, QR, and Cholesky ✓
- Solvers and matrix inversion ✓
- All BLAS 3 (fixed + variable) ✓
- SYMV, GEMV (fixed + variable) ✓

MAGMA 2.2 DRIVER ROUTINES

<table>
<thead>
<tr>
<th>MATRIX</th>
<th>OPERATION</th>
<th>ROUTINE</th>
<th>INTERFACES</th>
</tr>
</thead>
<tbody>
<tr>
<td>GE</td>
<td>Solve using LU</td>
<td>{sdcz}gesv</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Solve using MP</td>
<td>{zc,ds}gesv</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td>SPD/HPD</td>
<td>Solve using Cholesky</td>
<td>{sdcz}posv</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Solve using MP</td>
<td>{zc,ds}posv</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td>LEAST SQUARES</td>
<td>Solve using QR</td>
<td>{sdcz}geqs</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Solve using MP</td>
<td>{zc,ds}geqrsv</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td>SY/HE</td>
<td>Compute e-values,</td>
<td>{sdcz}geev</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>optionally e-vectors</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computes all e-values,</td>
<td>{sd}syevd</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>optionally e-vectors</td>
<td>{cz}heevd</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Range (D&amp;C)</td>
<td>{cz}heevdx</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Range (B&amp;I It.)</td>
<td>{cz}heevx</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Range (MRRR)</td>
<td>{cz}heevr</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td>STAND. SVP</td>
<td>Compute SVD,</td>
<td>{sdcz}gesvd</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>optionally s-vectors</td>
<td>{sdcz}gesdd</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td>GENERALIZED EVP</td>
<td>Compute all e-values,</td>
<td>{sd}sygvd</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>optionally e-vectors</td>
<td>{cz}hegvd</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Range (D&amp;C)</td>
<td>{cz}hegvdx</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Range (B&amp;I It.)</td>
<td>{cz}hegvx</td>
<td>CPU, GPU</td>
</tr>
<tr>
<td></td>
<td>Range (MRRR)</td>
<td>{cz}hegvr</td>
<td>CPU, GPU</td>
</tr>
</tbody>
</table>

ABBREVIATIONS

- GE: General
- SPD/HPD: Symmetric/Hermitian Positive Definite
- TR: Triangular
- D&C: Divide & Conquer
- B&I It: Bisection & Inverse Iteration
- MP: Mixed-precision Iterative Refinement

NAMING CONVENTION

magma_{routine name}_{gpu}

APPLICATIONS / LIBRARIES

Deep learning
- Structural mechanics
- Astrophysics
- Sparse direct solvers
- High-order FEM simulations

DEVICES

NVIDIA Pascal P100 GPU
- 56 MP x 64 @ 1.19 GHz

TEST MATRICES

from The University of Florida Sparse Matrix Collection http://www.cise.ufl.edu/research/sparse/matrices/

PERFORMANCE

Sparse matrix - vector product (SpMV) in double precision arithmetic

NVIDIA Pascal P100 GPU
- 56 MP x 64 @ 1.19 GHz

magma_{routine name}_{gpu}

TEST MATRICES

from The University of Florida Sparse Matrix Collection http://www.cise.ufl.edu/research/sparse/matrices/