MAGMA

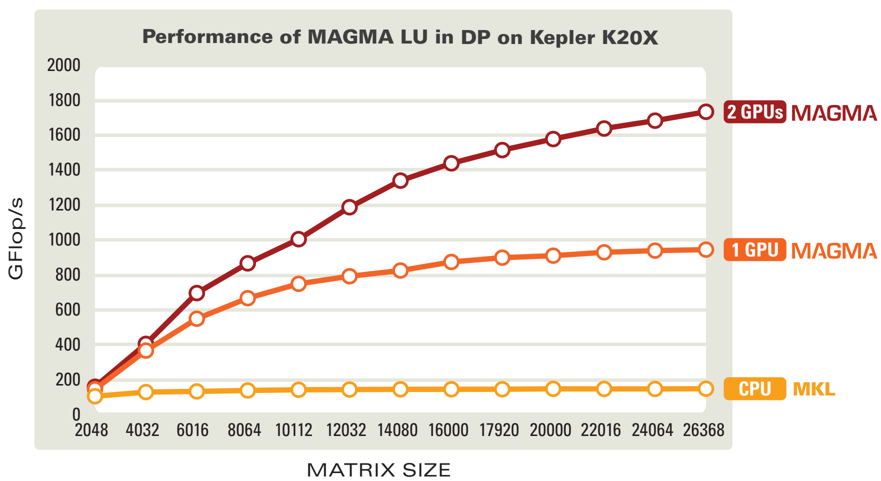
MATRIX ALGEBRA ON GPU AND MULTICORE ARCHITECTURES

MAGMA (Matrix Algebra on GPU and Multicore Architectures) is a collection of next generation linear algebra (LA) 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 (e.g. StarPU). Interfaces for the current LAPACK and BLAS standards are supported to allow computational scientists to effortlessly port any LA-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



GPU K20X
14 MP x192 @ 0.73 GHz

CPU Intel Xeon ES-2670
16 cores @ 2.60 GHz

FEATURES AND SUPPORT

- **MAGMA 1.3** FOR **CUDA**
- **clMAGMA 1.0** FOR **OpenCL**
- **MAGMA MIC 0.3** FOR **Intel Xeon Phi**

CUDA	OpenCL	Intel Xeon Phi	
●	●	●	Linear system solvers
●	●		Eigenvalue problem solvers
●			MAGMA BLAS
●			CPU Interface
●	●	●	GPU Interface
●	●	●	Multiple precision support
●			Non-GPU-resident factorizations
●			Multicore and multi-GPU support
●			Tile factorizations with StarPU dynamic scheduling
●	●	●	LAPACK testing
●	●	●	Linux
●			Windows
●			Mac OS

KEENELAND

MAGMA provides the scientific libraries that power **the Keeneland Project**, an NSF-funded partnership to enable large-scale computational science on heterogeneous architectures

FIND OUT MORE AT **http://keeneland.gatech.edu/**

SPONSORED BY

INNOVATIVE COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE

WITH SUPPORT FROM
AMD intel MathWorks Microsoft NVIDIA

IN COLLABORATION WITH
Berkeley University of Colorado Denver Inria KAUST

FIND OUT MORE AT
http://icl.utk.edu/magma

MAGMA

MATRIX ALGEBRA ON GPU AND MULTICORE ARCHITECTURES

MAGMA 1.3 ROUTINES & FUNCTIONALITIES	SINGLE GPU	MULTI-GPU STATIC	MULTI-GPU DYNAMIC
One-sided Factorizations (LU, QR, Cholesky)	✓	✓	✓
Linear System Solvers	✓		✓
Linear Least Squares (LLS) Solvers	✓		✓
Matrix Inversion	✓		✓
Singular Value Problem (SVP)	✓		
Non-symmetric Eigenvalue Problem	✓	✓	
Symmetric Eigenvalue Problem	✓	✓	
Generalized Symmetric Eigenvalue Problem	✓	✓	

SINGLE GPU
Hybrid LAPACK algorithms with static scheduling and LAPACK data layout

MULTI-GPU STATIC
Hybrid LAPACK algorithms with 1D block cyclic static scheduling and LAPACK data layout

MULTI-GPU DYNAMIC
Tile algorithms with StarPU scheduling and tile matrix layout

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]

DRIVER ROUTINES IN MAGMA 1.3

	MATRIX	OPERATION	ROUTINE	INTERFACES CPU	INTERFACES GPU
LINEAR EQUATIONS	GE	Solve using LU	{sdcz}gesv	✓	✓
		Solve using MP	{zc,ds}gesv		✓
	SPD/HPD	Solve using Cholesky	{sdcz}posv	✓	✓
		Solve using MP	{zc,ds}posv		✓
LLS	GE	Solve LLS using QR	{sdcz}geqrs		✓
		Solve using MP	{zc,ds}geqrsv		✓
STANDARD EVP	GE	Compute e-values, optionally e-vectors	{sdcz}geev	✓	
	SY/HE	Computes all e-values,	{sd}syevd	✓	✓
		optionally e-vectors	{cz}heevd	✓	✓
		Range (D&C)	{cz}heevdx		✓
		Range (B & I It.)	{cz}heevx	✓	✓
		Range (MRRR)	{cz}heevr	✓	✓
STAND. SVP	GE	Compute SVD,	{sdcz}gesvd	✓	
		optionally s-vectors		✓	
GENERALIZED EVP	SPD/HPD	Compute all e-values,	{sd}sygvd	✓	
		optionally e-vectors	{cz}hegvd	✓	
		Range (D&C)	{cz}hegvdx	✓	
		Range (B & I It.)	{cz}hegvx	✓	
		Range (MRRR)	{cz}hegvr	✓	

COMPUTATIONAL ROUTINES IN MAGMA 1.3

	MATRIX	OPERATION	ROUTINE	INTERFACES CPU	INTERFACES GPU
LINEAR EQUATIONS	GE	LU	{sdcz}getrf	✓	✓
		Solve	{sdcz}getrs		✓
		Invert	{sdcz}getri		✓
	SPD/HPD	Cholesky	{sdcz}potrf	✓	✓
		Solve	{sdcz}potrs		✓
		Invert	{sdcz}potri		✓
	TR	Invert	{sdcz}trtri	✓	
ORTHOGONAL FACTORIZATIONS	GE	QR	{sdcz}geqrf	✓	
		QR w/ pivoting	{sdcz}geqp3		✓
		Generate Q	{sd}orgqr	✓	✓
			{cz}ungqr	✓	✓
		Multiply matrix by Q	{sd}ormqr	✓	✓
			{cz}unmqr	✓	✓
		LQ factorization	{sdcz}gelqf	✓	✓
		QL factorization	{sdcz}geqlf	✓	
		Multiply matrix by Q	{sd}ormql	✓	✓
			{cz}unmql	✓	✓
STANDARD EVP	GE	Hessenberg reduction	{sdcz}gehrd	✓	
		Generate Q	{sd}orghr	✓	
			{cz}unghr	✓	
	SY/HE	Tridiagonalization	{sd}sytrd	✓	
			{cz}hetrd	✓	
		Generate Q	{sd}orgtr		✓
			{cz}ungtr		✓
		Multiply by Q	{sd}ormtr	✓	✓
			{cz}unmtr	✓	✓
SVD	GE	Bidiagonalization	{sdcz}gebrd	✓	
GENERALIZED EVP	SPD/HPD	Reduction to standard form	{sd}sygst	✓	✓
			{cz}hegst	✓	✓

SPONSORED BY

INNOVATIVE COMPUTING LABORATORY
THE UNIVERSITY of TENNESSEE

WITH SUPPORT FROM
AMD intel MathWorks Microsoft NVIDIA

IN COLLABORATION WITH
Berkeley University of California · University of Colorado Denver · Inria · KAUST

FIND OUT MORE AT
http://icl.utk.edu/magma

CITR CENTER FOR INFORMATION TECHNOLOGY RESEARCH