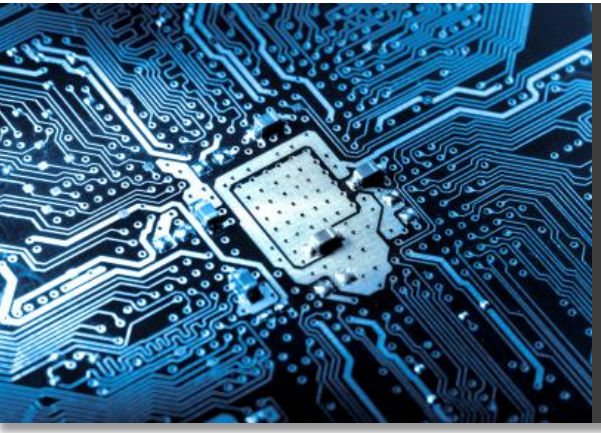


GTC 2012:

New Advances in GPU Linear Algebra

Kyle Spagnoli
EM Photonics

5/16/2012

Quick "About Us"

- HPC/GPU Consulting Firm
- Specializations in:
  - Electromagnetics
  - Image Processing
  - Fluid Dynamics
  - Linear Algebra

Introduction to our libraries

- **CULA Dense**
  - Linear algebra routines
- **CULA Sparse**
  - Iterative sparse system solvers and preconditioners
- **pCULA**
  - Scalable solvers for multiple GPUs
- Ongoing work

EM Photonics

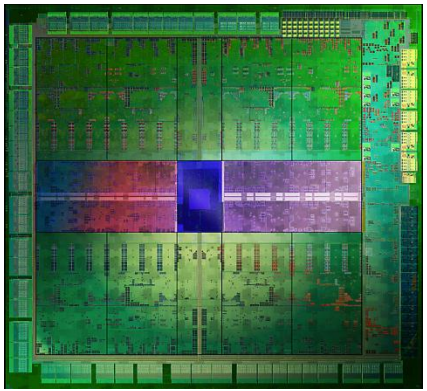
Introduction - Common Points

- Easy to use
  - No GPU programming experience necessary
  - `dgetrf(…)` → culaDgetrf(…)
- Exhaustively tested and benchmarked
  - Accuracy & stability first!
- Cross platform
  - Linux, Windows, Mac OS X
- Multiple languages
  - C/C++, Fortran, Python, Matlab

CULA Dense

CULA Dense – Introduction

- First released in 2009
- LAPACK and BLAS implementations
  - Host or device memory
  - Almost 300 routines
- Upcoming release (R15)
  - Tuned for Kepler architecture
  - Now free for personal academic use

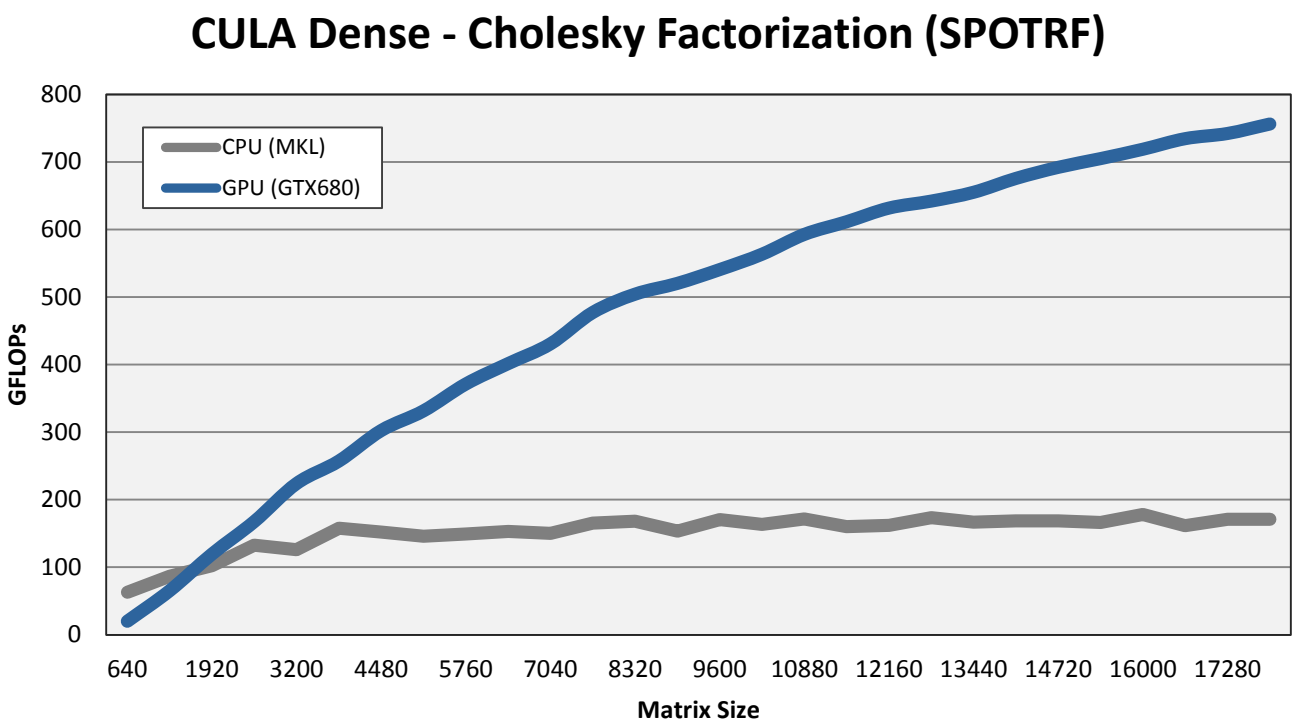


EM Photonics

CULA Dense - Functionality

LAPACK		BLAS
LU factorization	Cholesky factorization	Matrix-matrix multiply
QR decomposition	Orthogonal factorization	Matrix-vector multiply
Least squares	System solve	Rank updates
Eigenvalue routines	Matrix inversion	Conjugate
Singular value decomposition	Auxiliary routines	Transpose

CULA Dense - Performance



Performance numbers *include* transfer time across PCI-Express (Gen2) bus

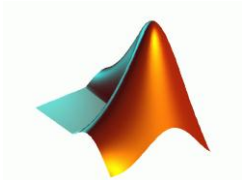
CPU
Intel Core i7 2600K

GPU
NVIDIA GTX 680 (1.5 GB)

EM Photonics

CULA Dense – Link Interface

- GPU acceleration with **no** code changes!
  - Intercepts calls to BLAS & LAPACK libraries
  - Analyze routine, parameters, and hardware
    - Forward to GPU if appropriate
    - Pass-through to CPU otherwise



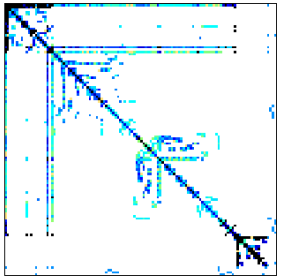
```
SET LAPACK_VERSION = cula_link.dll
SET BLAS_VERSION = cula_link.dll
```


EM Photonics

CULA Sparse (Iterative)

CULA Sparse – Introduction

- First released in 2011
- Iterative solvers and preconditioners
- Multiple matrix storage formats supported
- Upcoming release (S3)
  - Tuned for Kepler
  - Free for personal academic use



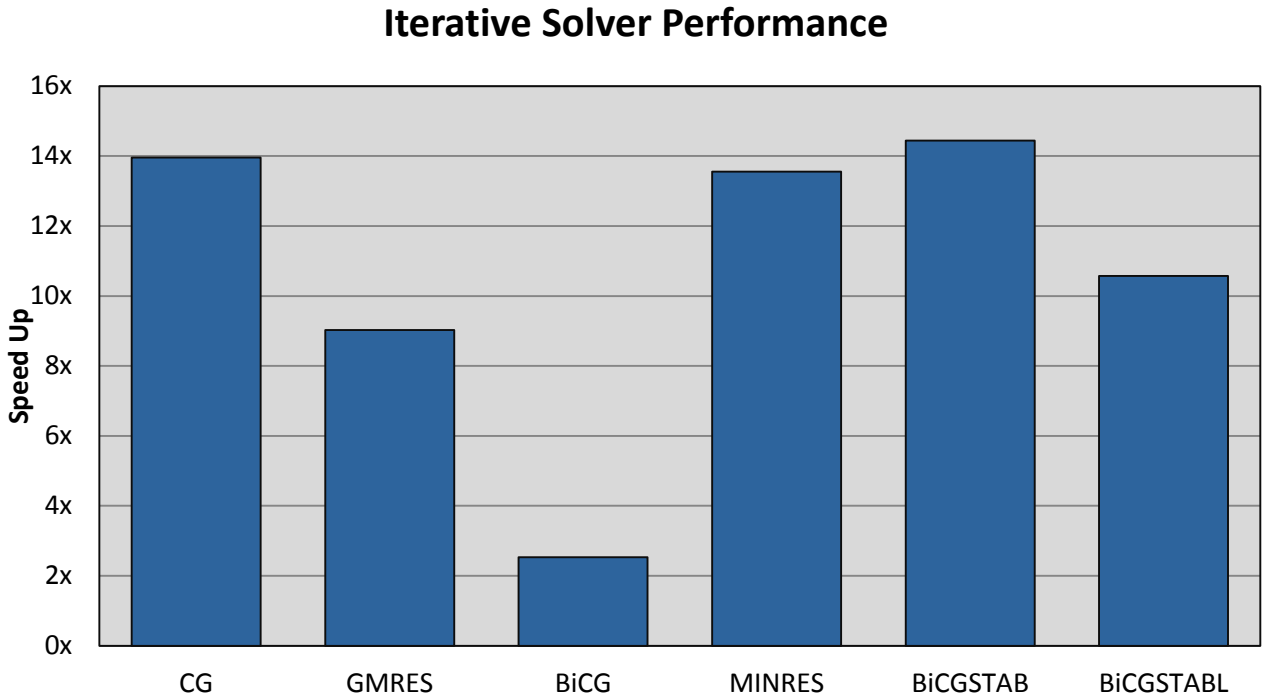
EM Photonics

CULA Sparse - Functionality

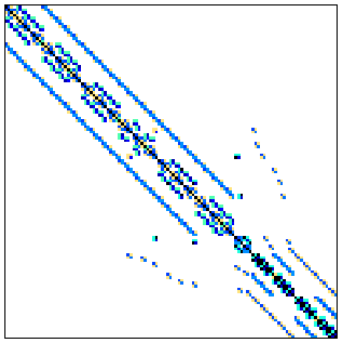
Solvers	Preconditioners	Data
CG	Jacobi	Double / Complex
BiCG	Block Jacobi	CSR / CSC / COO
BiCG-Stab / (L)	ILU0	
GMRES	Reordered ILU0	
MINRES		

EM Photonics

CULA Sparse - Performance



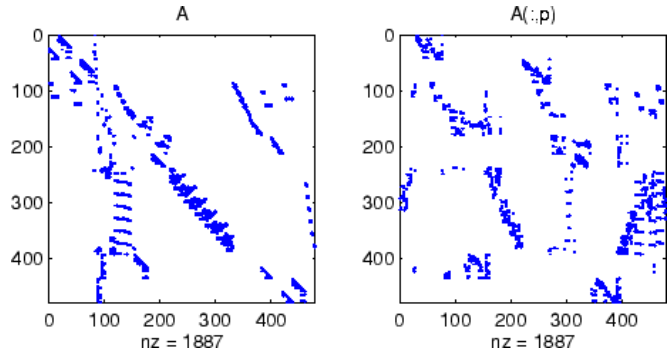
System Size = 1.5M
GPU = NVIDIA C2070
CPU = Xeon X5560 (MKL)



EM Photonics

CULA Sparse – Performance Features

- Hybrid performance
  - CPU begins working during initial transfer
    - Preconditioner generation
    - Initial iterations
- Matrix reordering
  - Can increase parallelism



EM Photonics

EM Photonics

Accelerated Computing Solutions

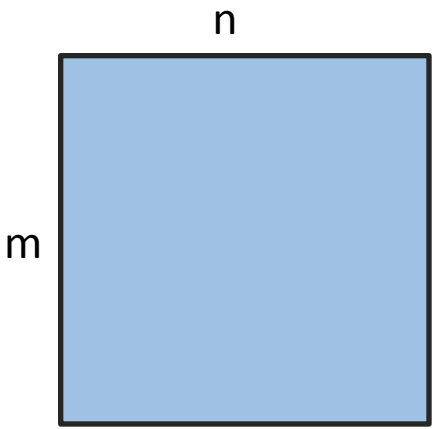
pCULA – Multi-GPU + CPU Performance

pCULA – Introduction

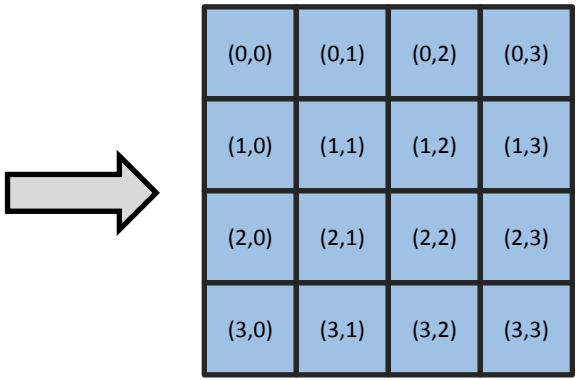
- Scale to multiple GPUs and CPUs in a single node
- Currently in alpha release
- Greatly increased performance, scalability, and functionality coming soon!

EM Photonics

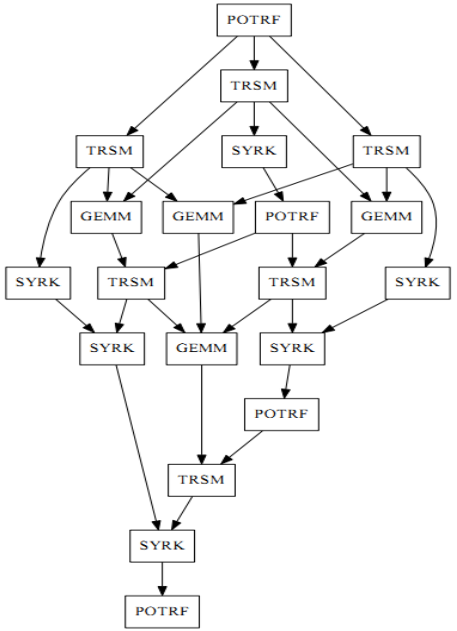
pCULA – Tiled Algorithms



original matrix

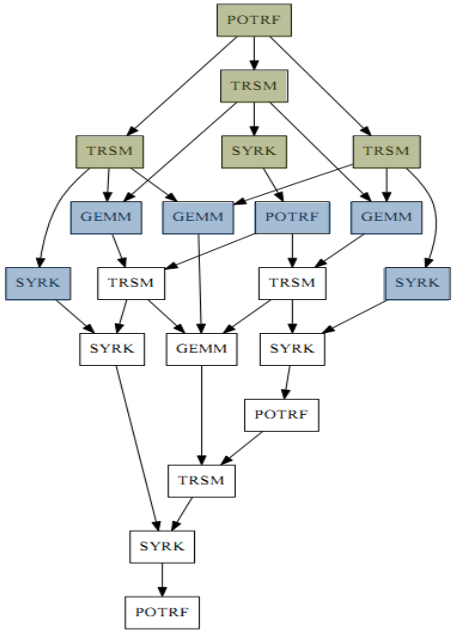


tiled matrix



EM Photonics

pCULA – Task Scheduling



Completed Tasks

- POTRF
- TRSM
- TRSM
- TRSM
- SYRK

Pending Valid Tasks

- GEMM
- GEMM
- GEMM
- POTRF
- SYRK
- SYRK

Hardware

Busy

(3 tasks)

Free

(1 task)

Free

(0 tasks)

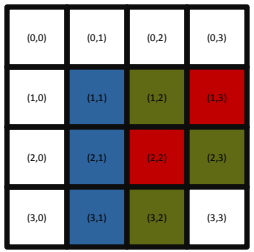
EM Photonics

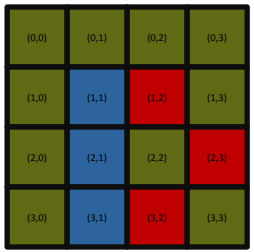
pCULA – Heterogeneous Task Scheduling

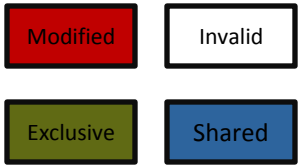
- Data locality is critical
- Hardware performance
  - Persistent “live tuning” performance database
- Task queue depth
  - Too long → idle hardware if not perfect
  - Too short → worker starvation

pCULA – Out of (GPU) core

- Solve problems larger than GPU memory
  - Natural extension of tiled data partitioning
  - MESI memory coherence protocol
  - Least recently used replacement strategy





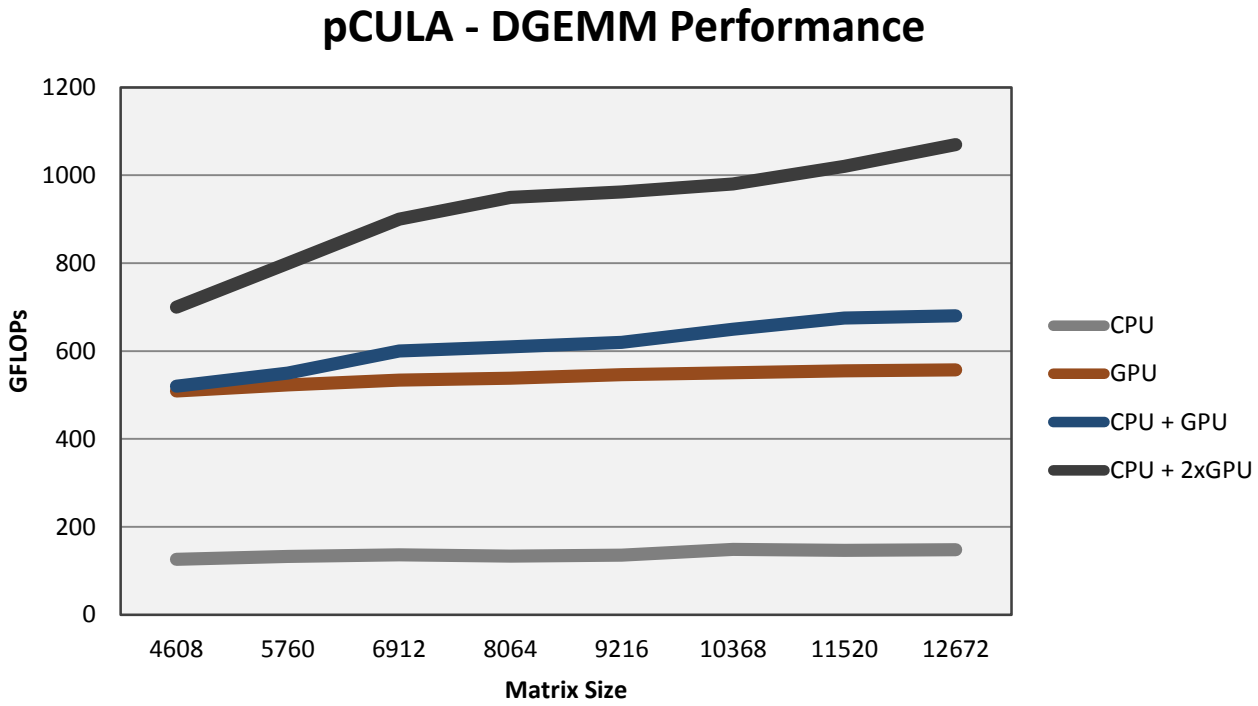


EM Photonics

pCULA – Function List

- Currently supports
  - BLAS Routines (GEMM, TRSM, GEMV)
  - LU Factorization & Solve (GETRF + GESV)
  - Cholesky Factorization & Solve (POTRF + POSV)
  - QR Factorization & Solve (GEQRF + GEQRS)
- Eigenvalue and SVD routines in future release

pCULA - Performance



Performance numbers *include* transfer time across PCI-Express (Gen2) bus

CPU
Intel Xeon 5560

GPU
2x NVIDIA C2050

EM Photonics

Ongoing Work

Ongoing Work - CULA

- CULA Dense
  - More routines/tuning
- CULA Sparse
  - Direct solvers
  - Algebraic Multi-Grid (AMG)
- pCULA
  - Multi-node cluster support
  - NUMA optimizations

Ongoing Work – C++ AMP

- Microsoft's C++ AMP library
  - "ampblas" development project
  - Linear algebra to C++ AMP ecosystem
  - Multiple talks today and tomorrow
  - C++ AMP Lounge

CULA Partners & Integrators

» Here at GTC 2012....

THANKS!

Thanks!
Questions?

- Convention hall @ booth #20
- More information @ www.culatools.com

EM Photonics