Scientific Libraries for Cray Systems Current Features and Future Plans

Mary Beth Hribar, Cray Inc.

Chip Freitag, AMD

CRAY USER GROUP INCORPORATED

Overview

- Scientific libraries for Cray systems (Mary Beth)
  - Cray X1 series
  - Cray XT3
  - Cray XD1
- LibSci (Mary Beth)
- ACML (Chip)

Cray Contributors

- Mary Beth Hribar, Manager
- Adrian Tate, ScaLAPACK
- Bracy Elton, FFTs
- Chao Yang, BLAS, LAPACK, sparse solvers
- John Lewis, ScaLAPACK/LAPACK, sparse solvers
- Neal Gaarder, libm

AMD Contributors

- Chip Freitag, Member of Technical Staff
- Tim Wilkens, Member of Technical Staff
- Preeta Raman, Strategic Alliance Manager - Software Development Tools, Segment and Industry Solutions

Cray's Family of Supercomputers

Cray X1E

- 1 – 50+ TFLOPS
- 16 – 8,138 processors
- Vector processing for uncompromised sustained performance

Cray XT3

- 1 – 50+ TFLOPS
- 256 – 30,000 processors
- MPP Compute system for large-scale sustained performance

Cray XD1

- 50 GFLOPS - 2+ TFLOPS
- 12 – 576+ processors
- Entry/Mid range system optimized for sustained performance
- With reconfigurable computing capability

Purpose-Built High Performance Computers

The Scientific Libraries

- Cray X1 series
  - LibSci
- Cray XT3
  - ACML
  - Cray XT3 LibSci
- Cray XD1
  - ACML
  - ScaLAPACK

Cray X1 Series

- LibSci provides
  - BLAS
    - OpenMP version of level 3, some level 2
    - Inline level 1, some level 2 with **`-O inlinelib`**
  - LAPACK
  - FFTs
    - Single processor
    - Distributed memory parallel
  - ScaLAPACK, BLACS
  - Sparse solvers (single precision only)

Cray X1 Series

- LibSci supports
  - 32- and 64-bit default data types
  - MSP and SSP modes
  - Serial and parallel programming models:
    - Single processor
    - 4-way (MSP mode) or 16-way (SSP mode) shared memory parallelism in level 3 BLAS
    - Distributed memory parallelism

Cray X1 Series Software Releases

- PE 5.4 (March 2005)
  - LAPACK built with level 1 BLAS inlined
  - Radix 7, 11, 13 butterflies for complex-to-complex FFTs
- PE 5.5 (December 2005)
  - Improved parallel LU (psgetrf, pdgetrf, pcgetrf, pzgetrf)
  - FFT improvements TBD

Cray XT3

- ACML
  - 64-bit libraries
  - GNU and PGI versions
- Cray XT3 LibSci
  - ScaLAPACK
  - BLACS
  - SuperLU_DIST
- Module environment similar to Cray X1

Cray XT3 Software Releases

1.1 June 2005	1.2 Sept 2005	2.0 Dec 2005
ACML 2.5 Cray XT3 LibSci: ScaLAPACK SuperLU_DIST	ACML 2.6	ACML 3.0 Cray FFT interface

Cray XD1

- ACML
  - 32- and 64-bit versions
  - GNU and PGI versions
- ScaLAPACK, BLACS
  - In /usr/local/lib64
  - Use with PGI 6.x compilers
- Library modules not part of software release (yet)

Cray XD1 Software Releases

1.2 May 2005	1.3 Aug 2005	1.4
ACML 2.5 ScaLAPACK BLACS	ACML 2.6 Library modules	ACML 3.0 Cray FFT interface Improvements in ScaLAPACK

LibSci Projects

- Cray FFT enhancements
- Cray FFTs on Cray XD1 and Cray XT3
- ScaLAPACK tuning
- Sparse solvers for Cray systems

Cray FFT Enhancements

- Special butterflies
  - Complex-to-complex case
  - Composite: 6, 10, 12, 15, 18, 20
  - Higher powers of 2 and 3 radices: 9, 16
  - Radices: 7, 11, 13
  - Reduce twiddle factor multiplication
  - Reduce memory traffic
- Better cache blocking

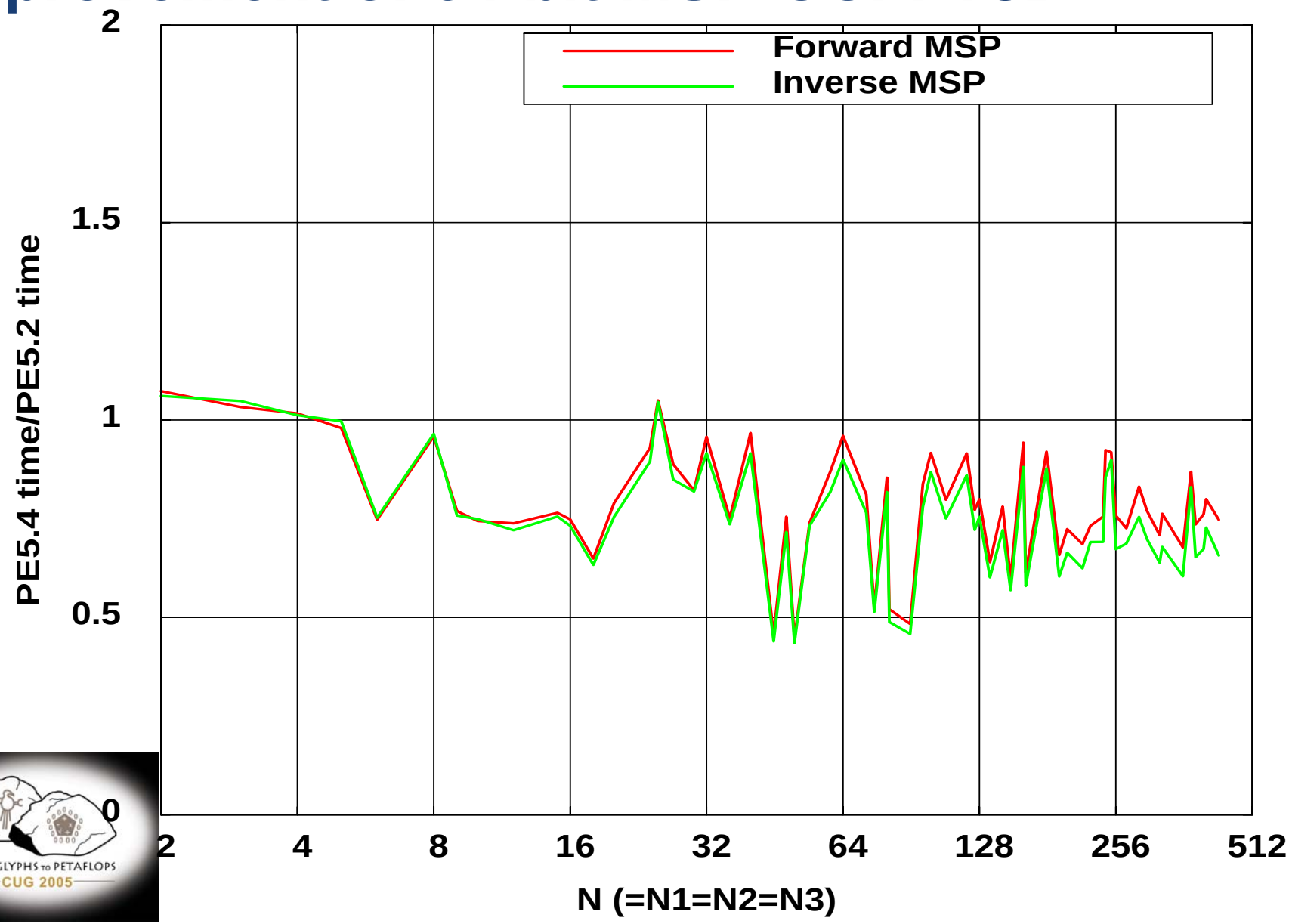
Cray FFT Enhancements

- Distributed memory parallel FFTs now contain two more (optional) workspace arguments
  - User can manage memory
  - Or associated workspace is allocated and deallocated within routines

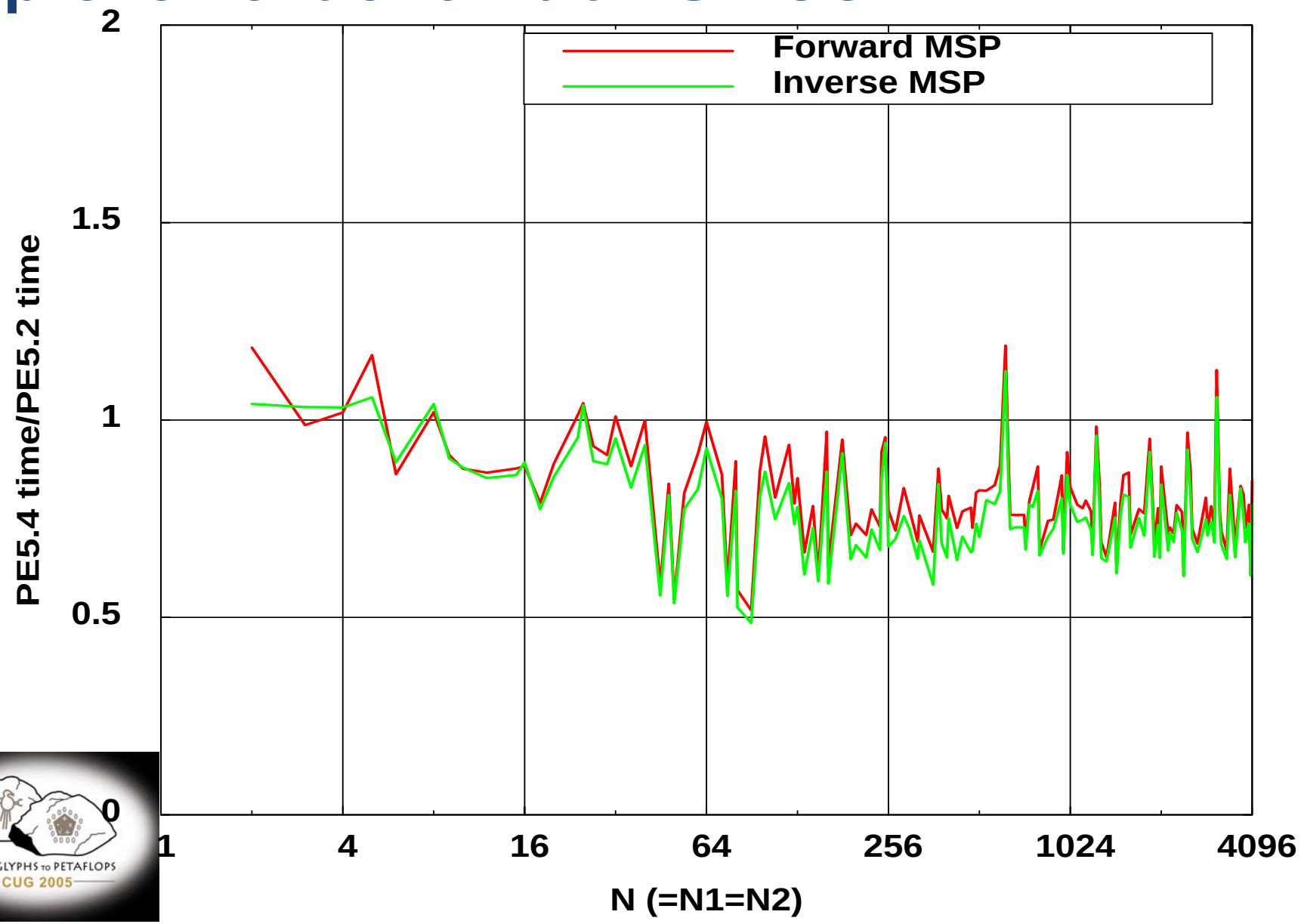
Performance of FFTs on Cray X1

- Compare FFTs in PE 5.2 (April 2004) and PE 5.4 (March, 2005)
- Plot ratio of PE 5.4 time to PE 5.2 time
  - Values less than 1 indicate improvement
- Results given for N=N1=N2=N3=M
- FFT length factors are powers of 2, 3, 5
- Leading dimensions yield odd multiples of 4 strides
- Performance tests run on single Cray X1 MSP
- Jaggedness: not all FFT lengths have factorizations that can take advantage of the new butterflies

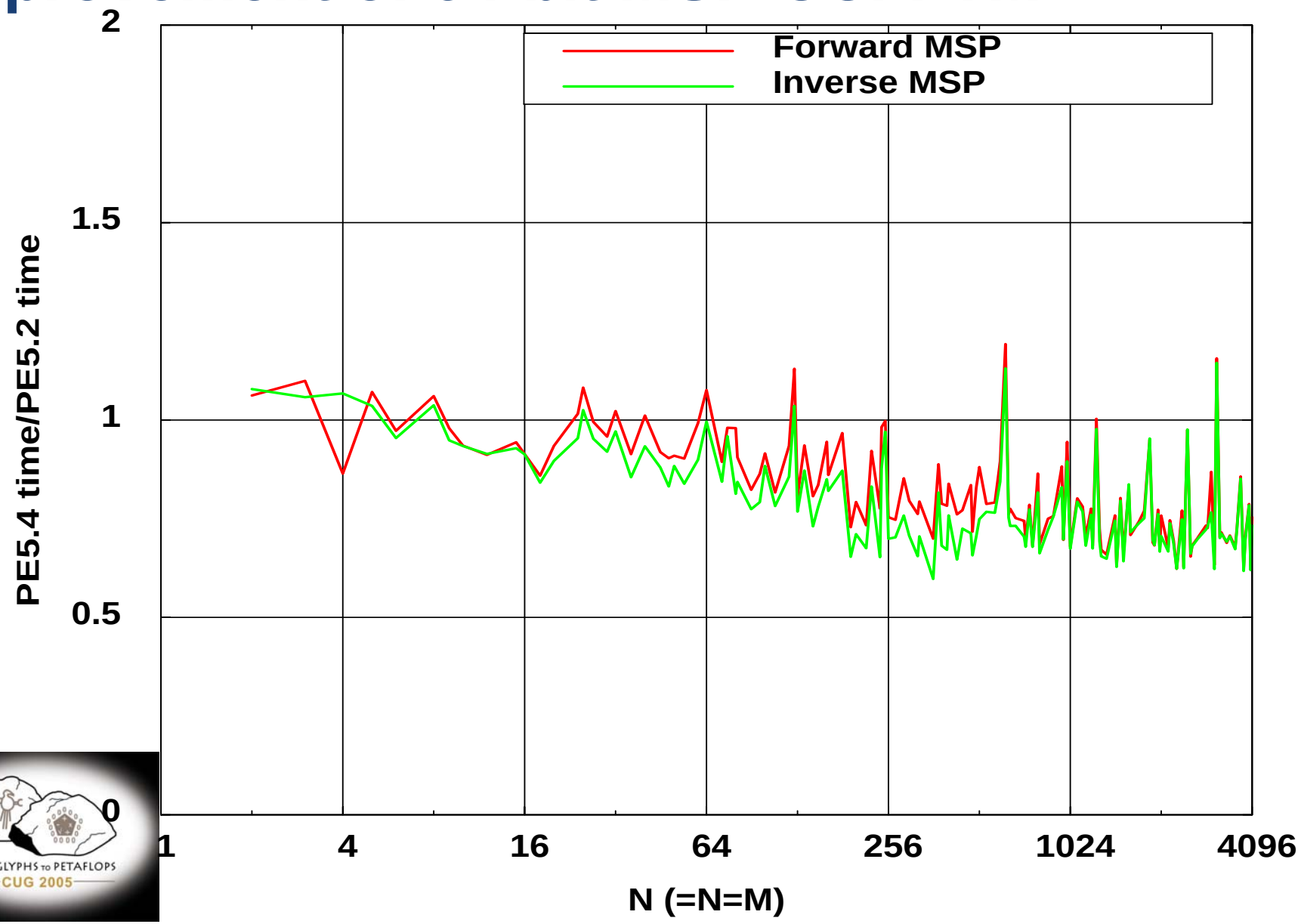
Improvement of 64-bit MSP CCFFT3D



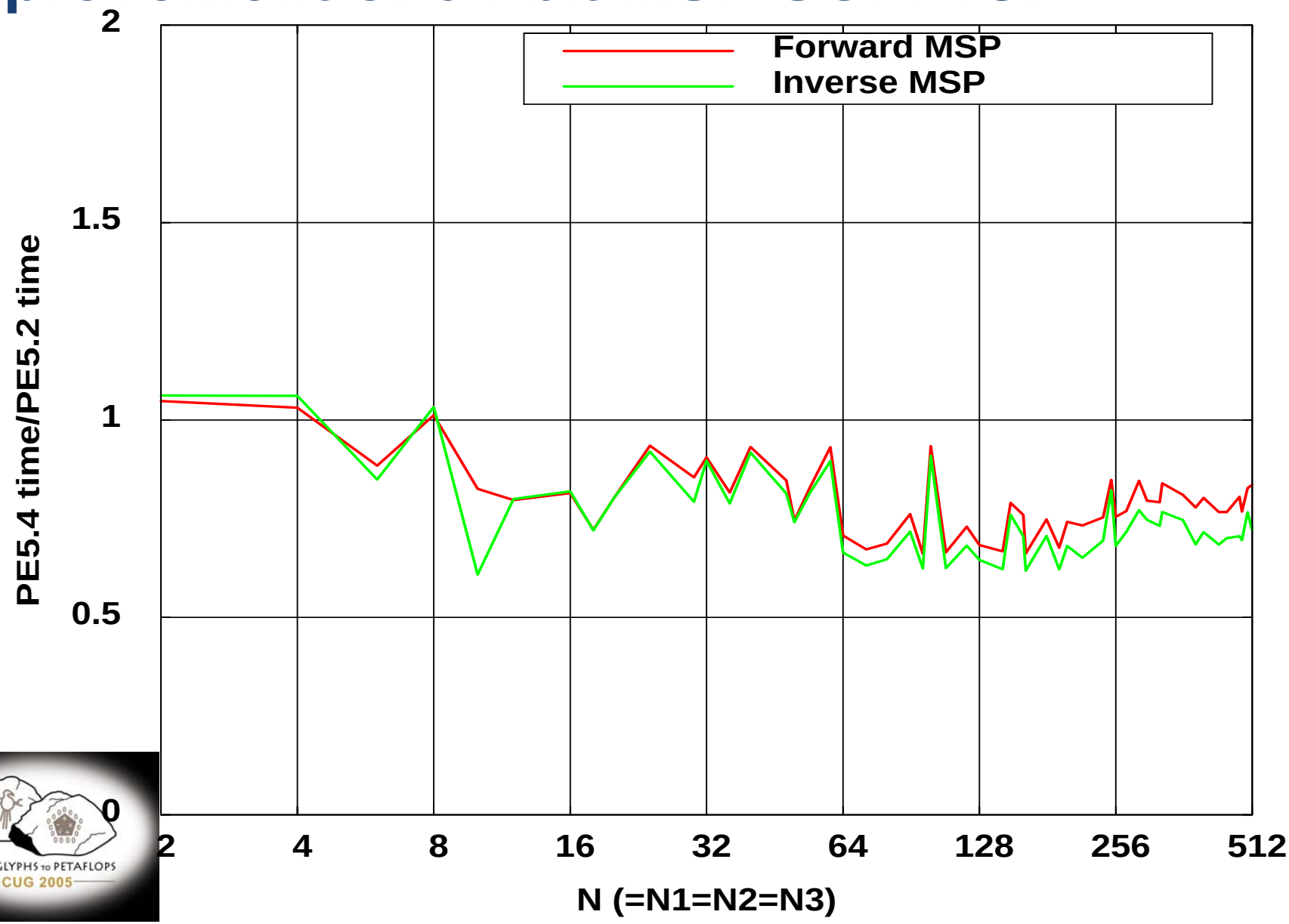
Improvement of 64-bit MSP CCFFT2D



Improvement of 64-bit MSP CCFFTM



Improvement of 64-bit MSP SCFFT3D



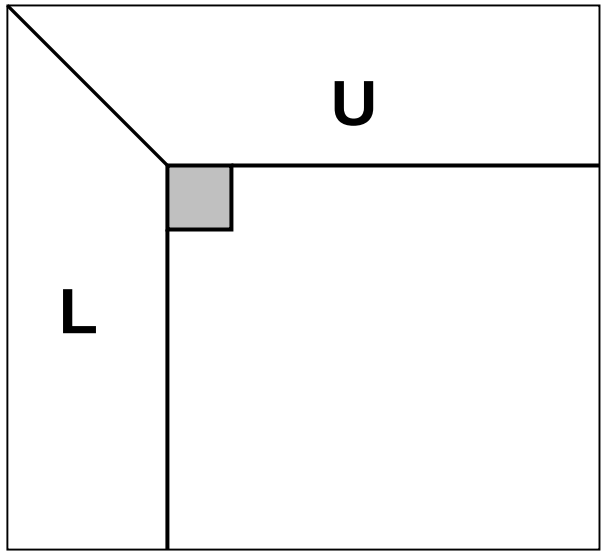
Cray FFTs on Cray XD1, Cray XT3

- Consistent FFT interface on all Cray systems
- Add to functionality in ACML
- Leverage ACML FFTs where possible & sensible
- Need access to lower level ACML routines
- Provide DMP FFTs also

ScaLAPACK Tuning

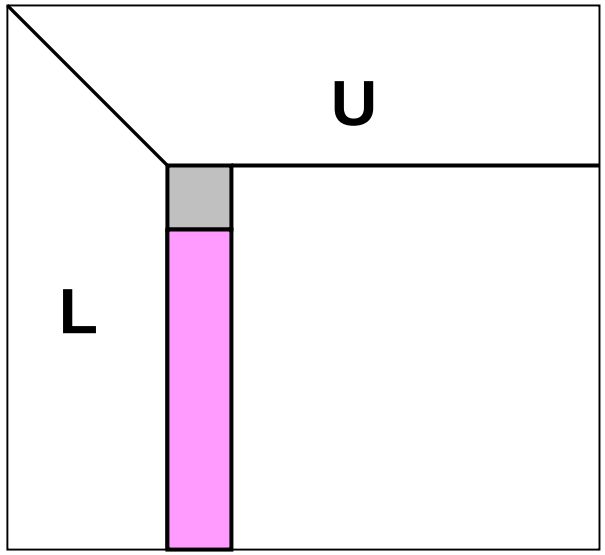
- Provide ScaLAPACK on all systems
- Tune for Cray's interconnect and message passing libraries
- Improve ScaLAPACK for any platform
  - Rewrite parallel algorithms to overlap computation and communication
  - Use one-sided communication
  - Add new BLACS routines
- Joint work with Osni Marques, Tony Drummond at NERSC

Snapshot of Block LU



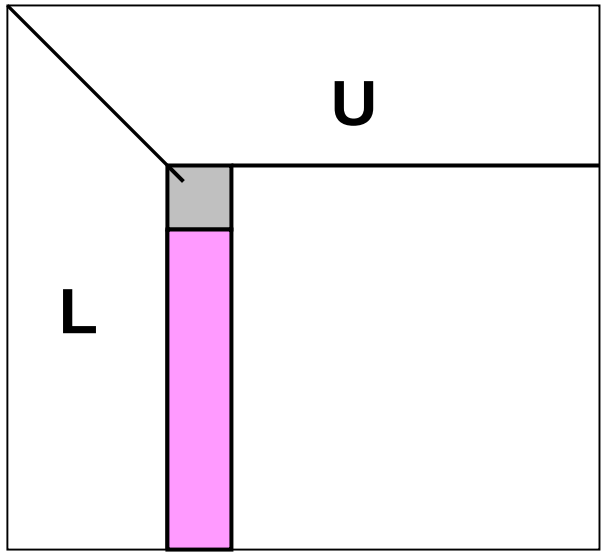
diagonal block

Snapshot of Block LU



current ‘panel’

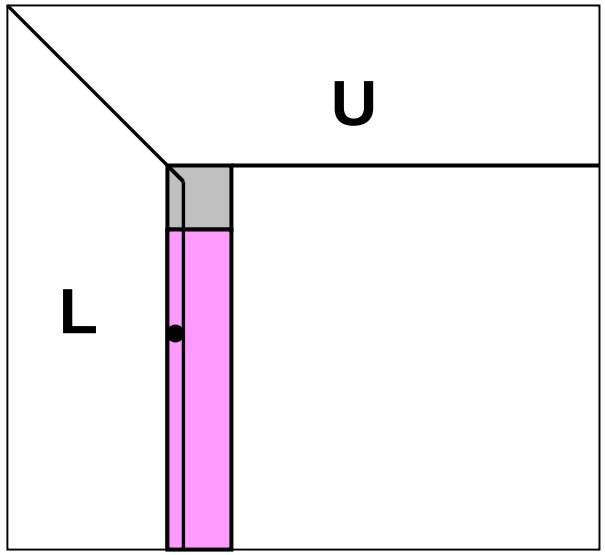
Snapshot of Block LU



begin factorization of diagonal block

(pdgetf2)

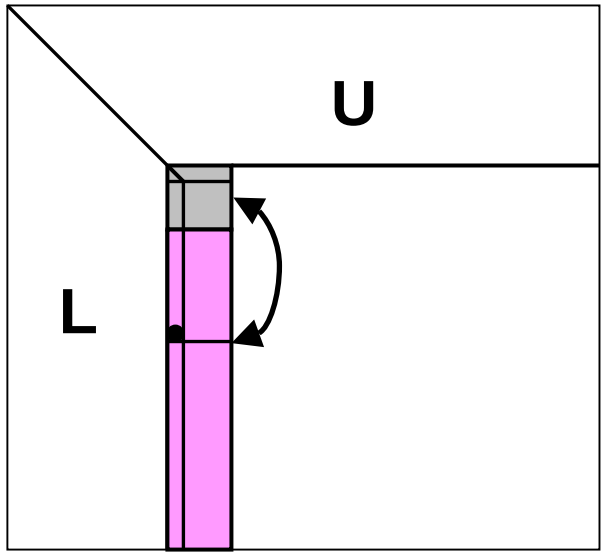
Snapshot of Block LU



Find maximum element in current column (pivot)

(pdamax)

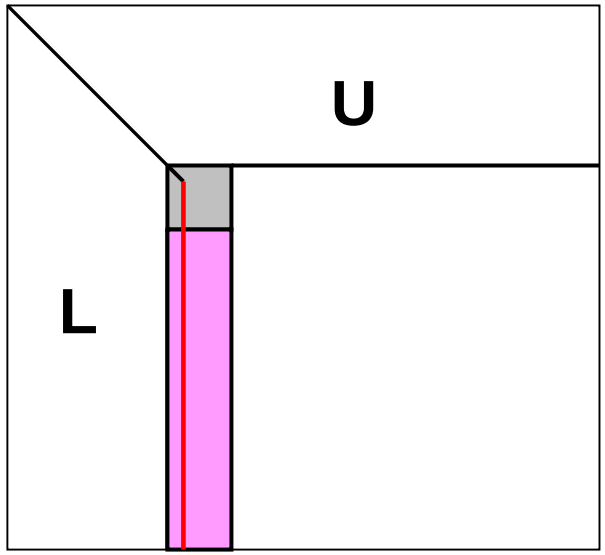
Snapshot of Block LU



Swap rows within the current panel

(pdswap)

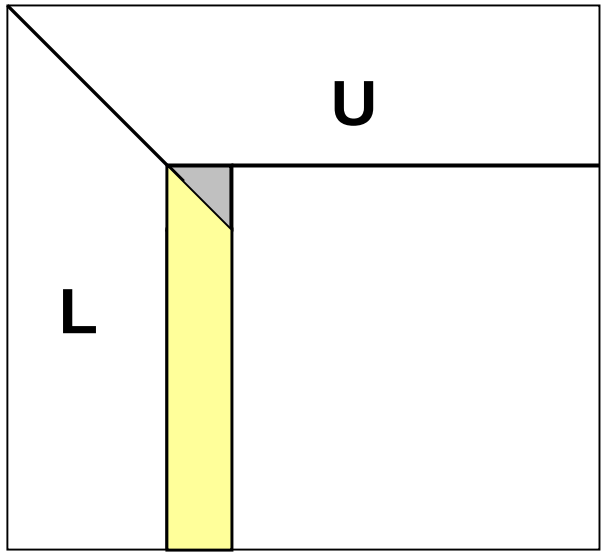
Snapshot of Block LU



Scale current column by pivot

(pdscal)

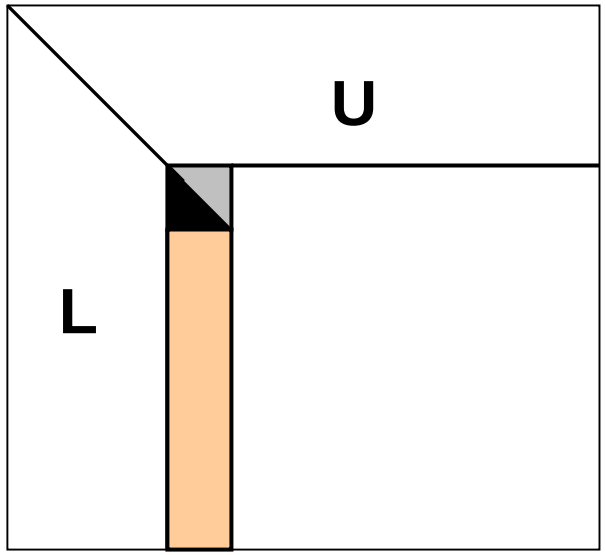
Snapshot of Block LU



Perform triangular solve

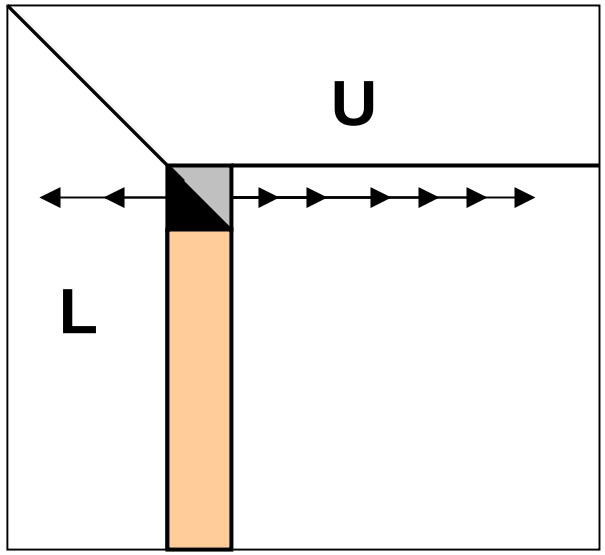
(pdger)

Snapshot of Block LU



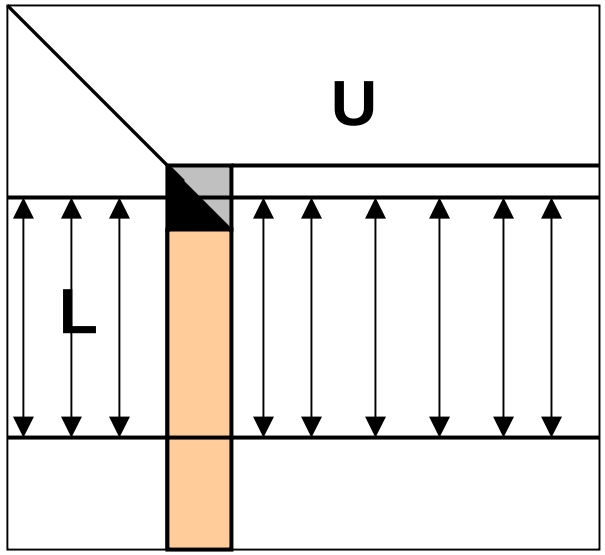
Repeat until block is factorized

Snapshot of Block LU



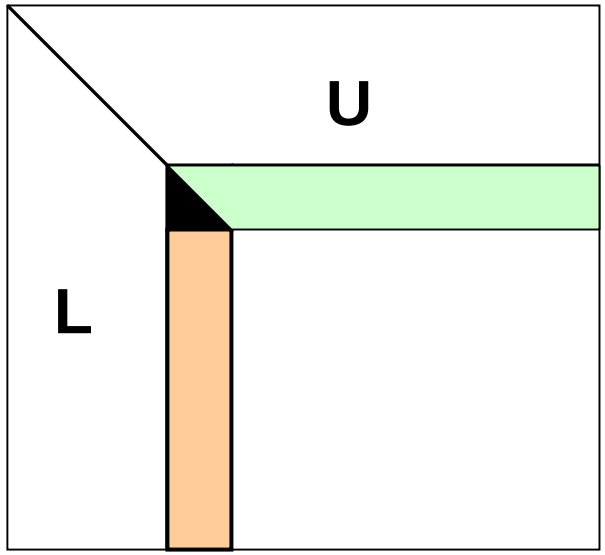
Broadcast pivot information

Snapshot of Block LU



Pivots applied to other column panels

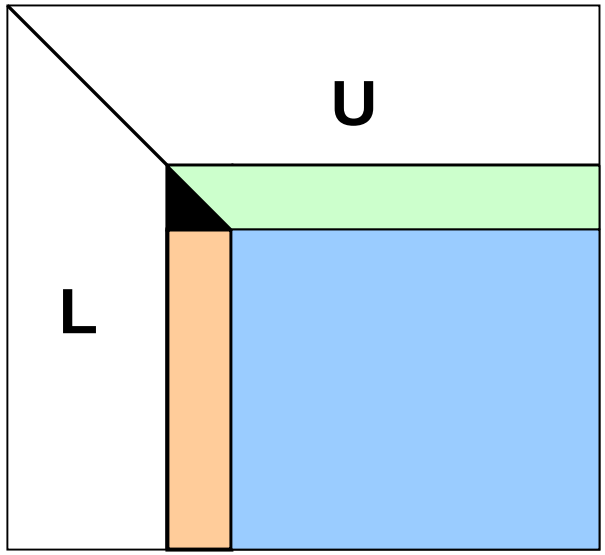
Snapshot of Block LU



Compute block row of U

(pdtrsm)

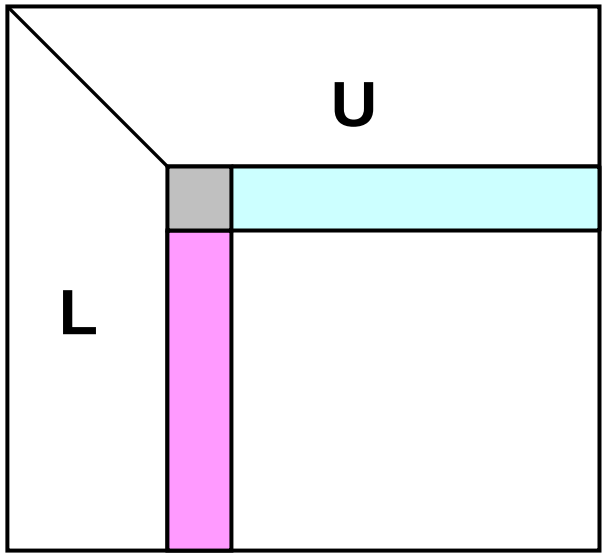
Snapshot of Block LU

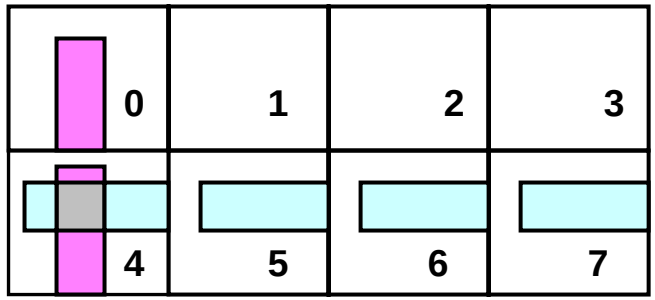


Update trailing submatrix

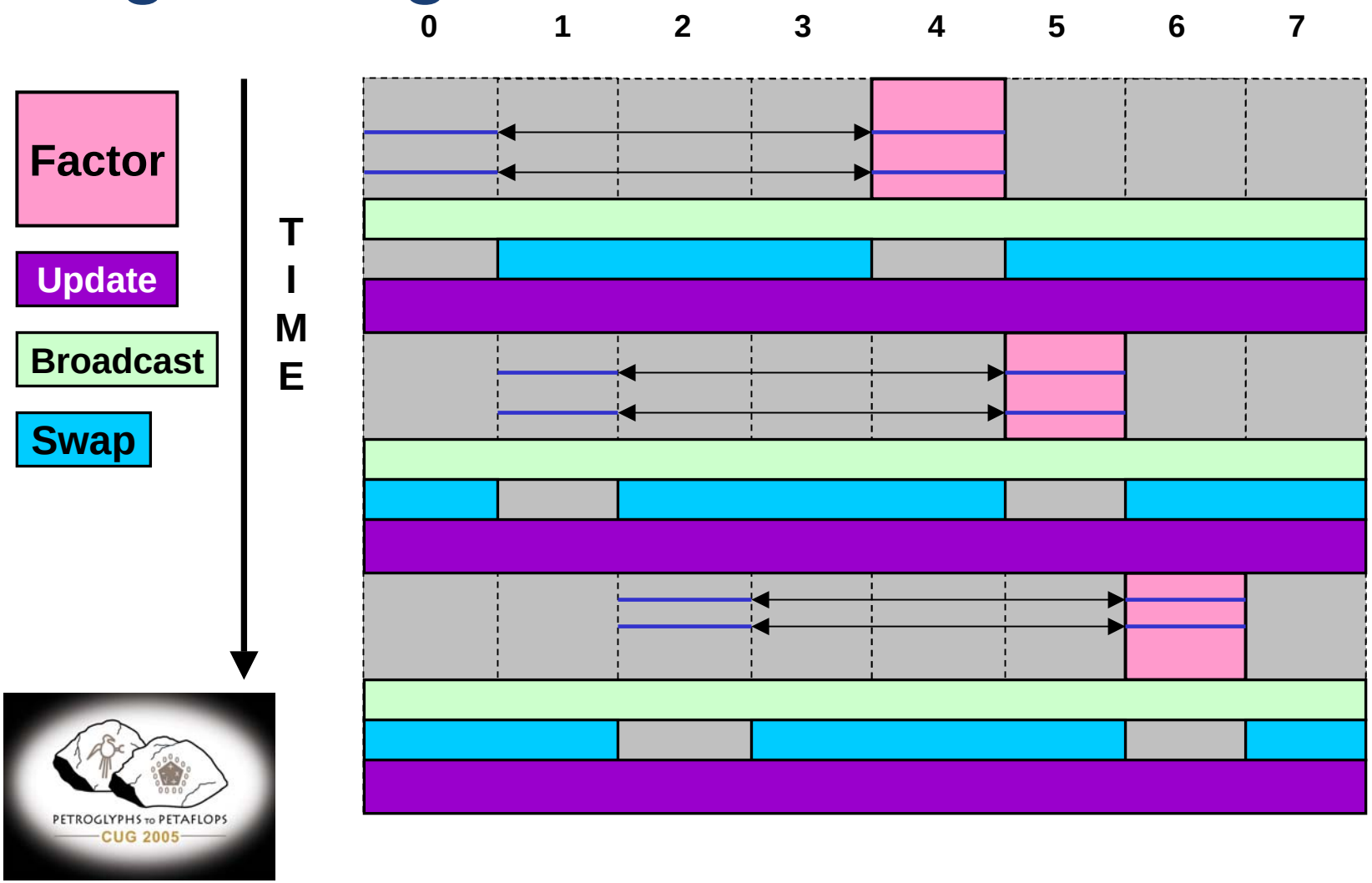
(pdgemm)

Map matrix to 2x4 grid





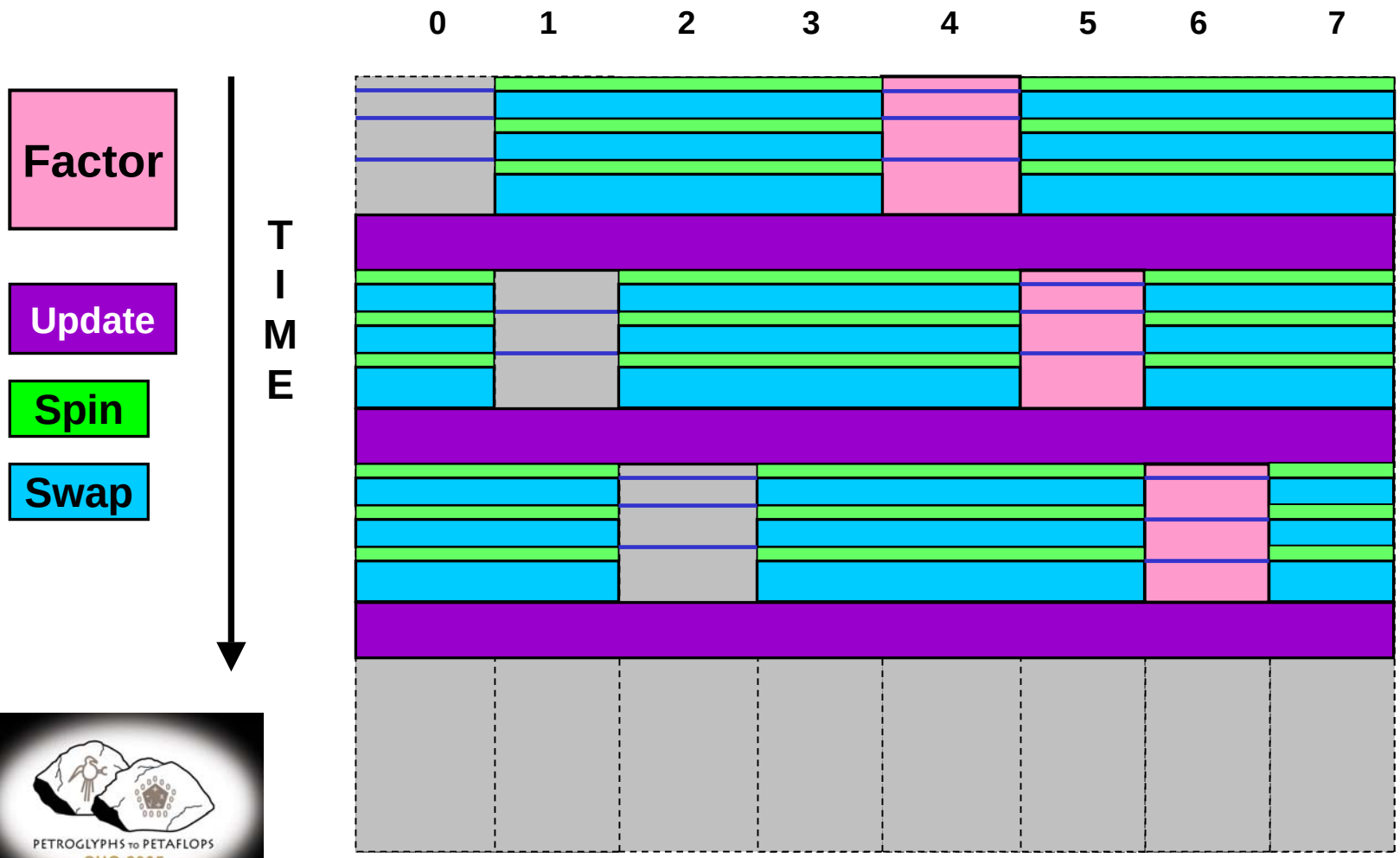
Timeline of next three iterations, original algorithm



Blacs_Spin

- New BLACS routine
- Processor queries information on remote processor
- Remote processor not involved
- One-sided communication using shmem
- In block LU, blacs_spin
  - Is used to check for pivot information
  - Allows idle processors to pivot during factorization

Timeline of next three iterations, new algorithm



Threshold pivoting

- Avoid pivoting to further reduce communication
- Find local pivot: maximum column entry on local processor
- Find global pivot: maximum column entry (traditional pivot)
- Compare diagonal element, local pivot and global pivot
  - » If diagonal element is large enough, do not pivot
  - » If local pivot is large enough, use local pivot

Performance of Improved LU

- Early tests show that we can expect a 20% improvement for large enough problem sizes
- Performing more experiments with threshold pivoting

Sparse Solvers

- SuperLU on future Cray systems
- Sparse matrix-vector multiply

Summary

- Work closely with AMD to provide library solution for all Opteron-based Cray platforms
- Leverage tuning work across multiple systems
- Collaborations with universities and labs
  - Advance development
  - Provide early access to new software