Hessenberg Reduction

with

Transient Error Resilience

on

GPU-Based Hybrid Architectures

Yulu Jia, Piotr Luszczek (presenting), Jack Dongarra

University of Tennessee, Knoxville Innovative Computing Laboratory

From a Single Accelerated Node to Exascale

- MTTF of the entire machine depends on reliability of each node
- The MTTF of the entire machine can be statistically computed based on single-node reliability for a number of distributions
  - Exp(1/100)
  - Weibull(0.7, 1/100)
  - Weibull(0.5, 1/100)
- See Yves Robert's work for detailed analysis

One node ~10^3 cores	MTTF = 1 year	MTTF = 10 years	MTTF = 120 years
↓	↓	↓	↓
Exascale machine ~10^6 nodes	30 seconds	5 minutes	1 hour

Field Data on Resilience

- Soft errors...
  - are caused by: cosmic rays (alpha particles, high energy and/or thermal neutrons)
  - occur in practice
    - Commercial study in 2000 by Sun Microsystems
    - ASC Q supercomputer at Los Alamos in 2003
    - Jaguar (Cray XT5) at ORNL
      - Nearly 225k cores
      - 1253 separate node crashes during 537 days (Aug 2008-Feb 2010)
      - Or 2.33 failures per day
      - Or less 10 hours of failure-free operation
    - … and any non-ECC machine
- Accelerators are common
  - In many shared-memory systems
  - Supercomputers
    - Tianhe-1A, Titan (Cray XK7, 560k cores), Tianhe-2 (3M+ cores)
- And at Exascale ~1 billion threads and MTTF < 1 day!

Hessenberg Reduction (HRD) and Its Applications

- HRD = Hessenberg Reduction
  - General (non-symmetric) eigenvalue problem
  - Generalized eigenvalue problem
- Applications
  - Structural mechanics
  - Spectral graph analysis
  - Control theory
  - …
- Complexity
  - $^{10}/_{3}\, n^3 + O(n^2)$
  - Both compute bound and memory bound

Numerical Eigenvalue Algorithm Recap

- To solve

  $Ax = \lambda x$ or $Ax = \lambda Bx$

- We transform matrix A into Hessenberg matrix H with the same eigenvalues:

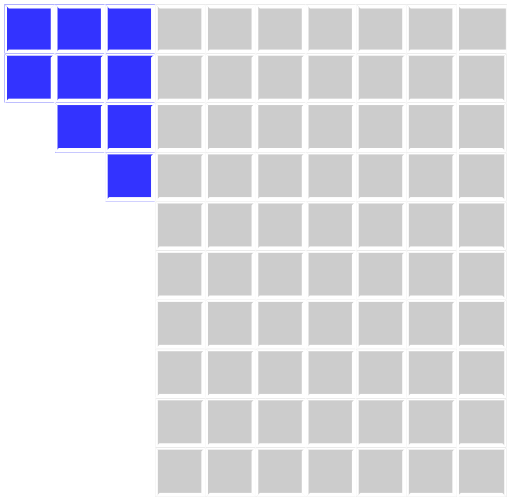
$$H_1 = Q_1^{\mathsf{T}} A\, Q_1$$
$$H_2 = Q_2^{\mathsf{T}} H_1\, Q_2$$
$$\dots$$
$$H_n = Q_n^{\mathsf{T}} H_{n-1}\, Q_n \equiv H$$

*	*	*	*	*	*
*	*	*	*	*	*
0	*	*	*	*	*
0	0	*	*	*	*
0	0	0	*	*	*
0	0	0	0	*	*

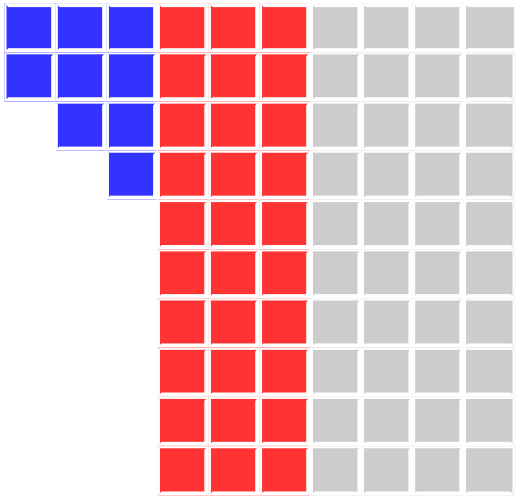
- H is in Hessenberg form:

- Iterative algorithm is used to find eigenvalues of H

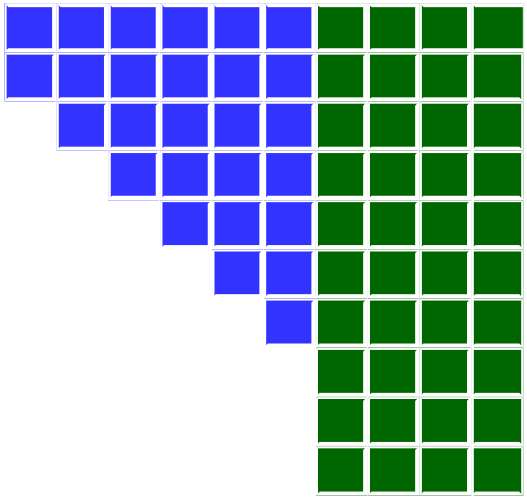
Panel-Update Approach



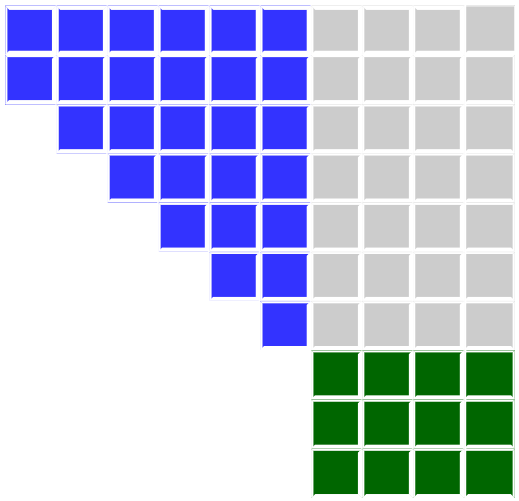
Begin iteration 1



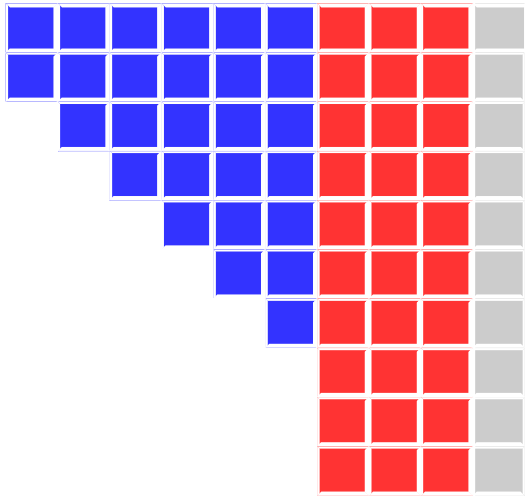
Panel 2



Right Update 2

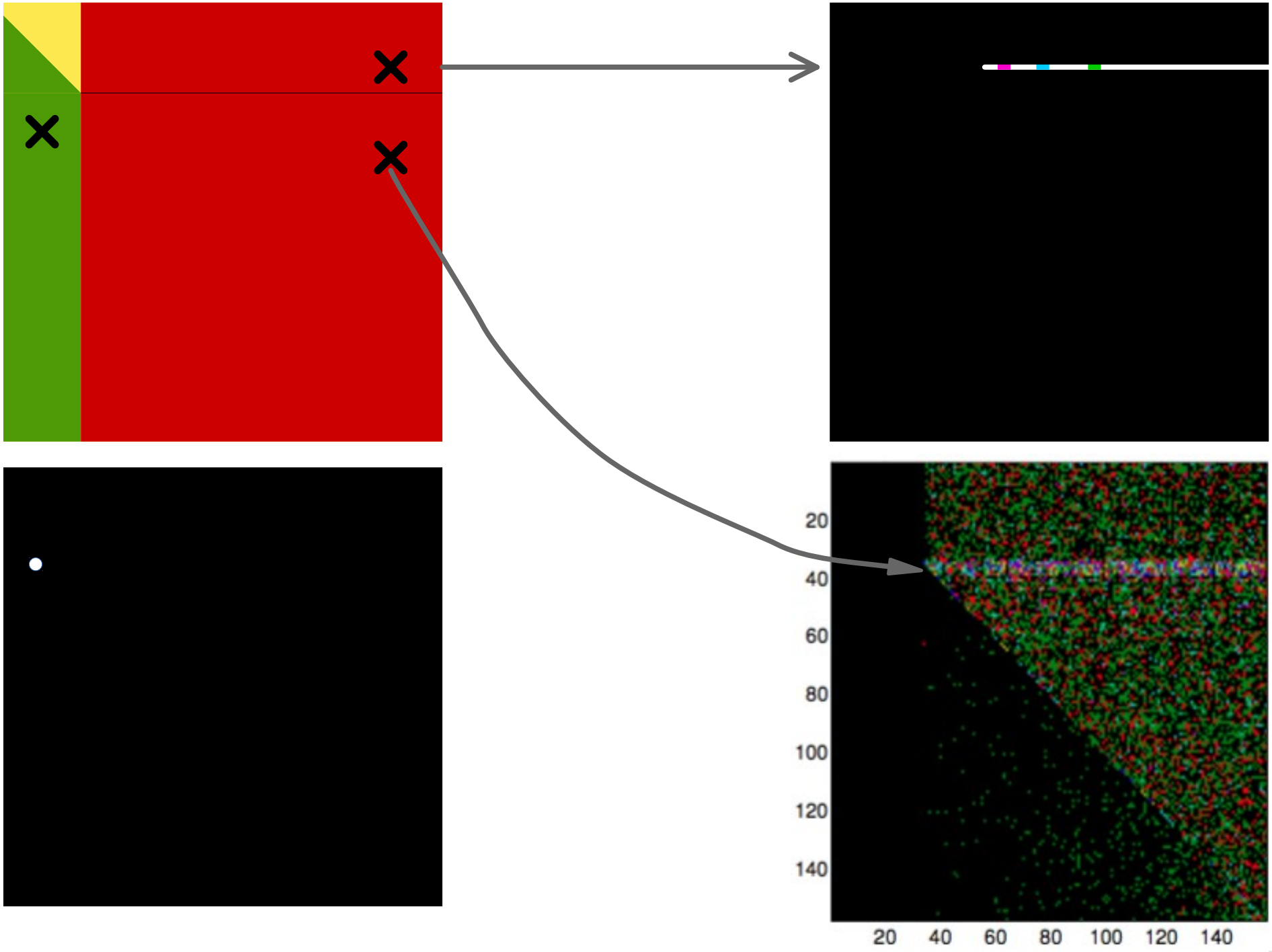


Left Update 2



Panel 3

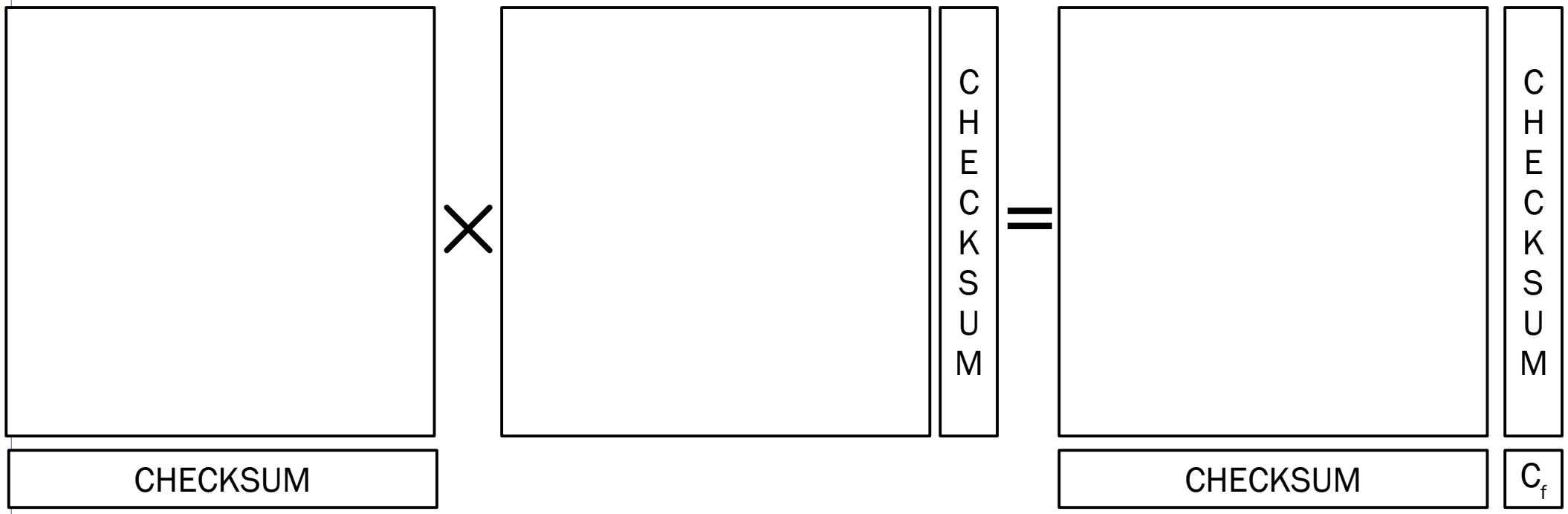
Propagation of Error During Hessenberg Reduction



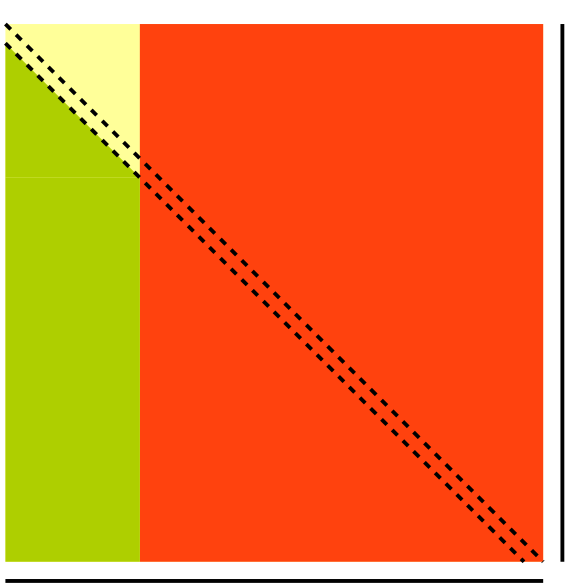
Techniques for Error Protection and Failure Recovery

- Algorithm-Based Fault Tolerance
  - Kuang-Hua Hua, Jacob Abraham, ABFT for Matrix Operations
    - Implementation on systolic arrays
  - Takes advantage of additional mathematical relationship(s)
    - Already present in algorithm
    - Introduced (cheaply, if possible) by ABFT – usually weighted sums
- Diskless checkpointing
  - Additional (small) data is kept in live processes or extra memory
  - No need for full I/O checkpointing

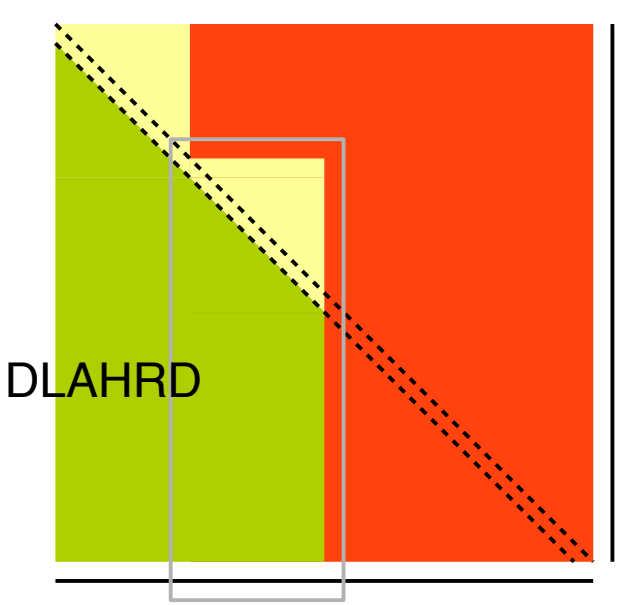
From Huang and Abraham: Checksum Mat-Mat-Mul



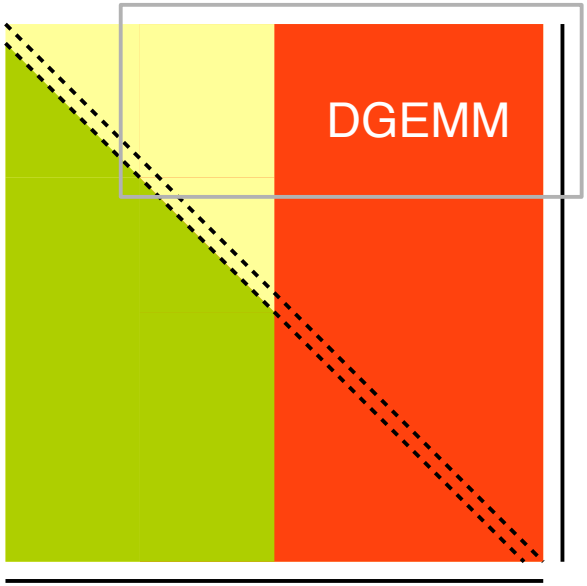
Fault Tolerant Hessenberg Reduction



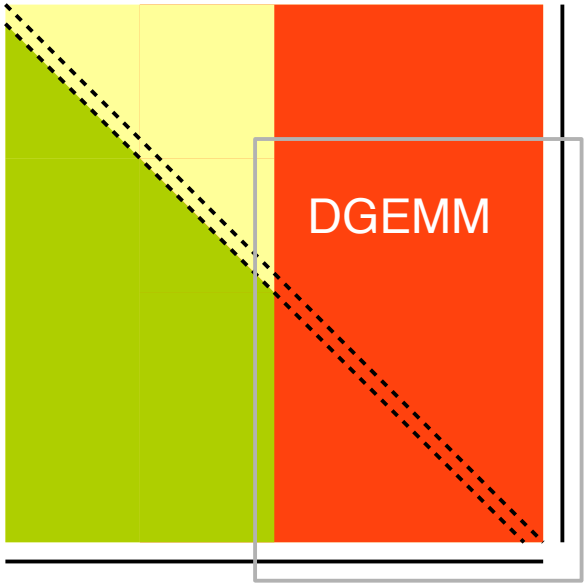
Begin iteration



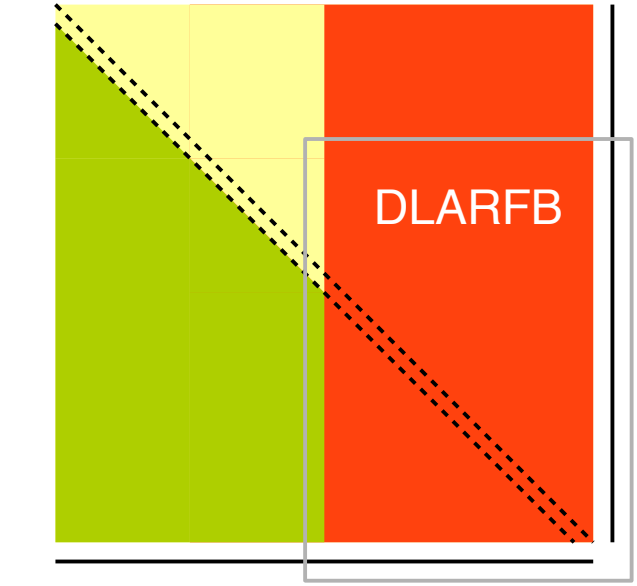
Factorize panel



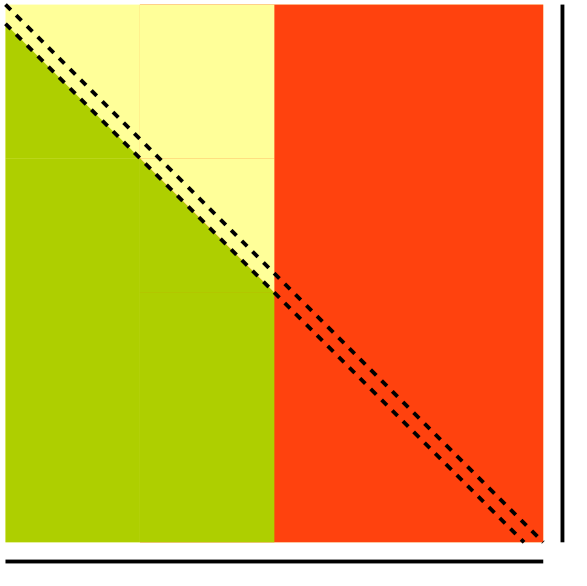
Right update



Left update

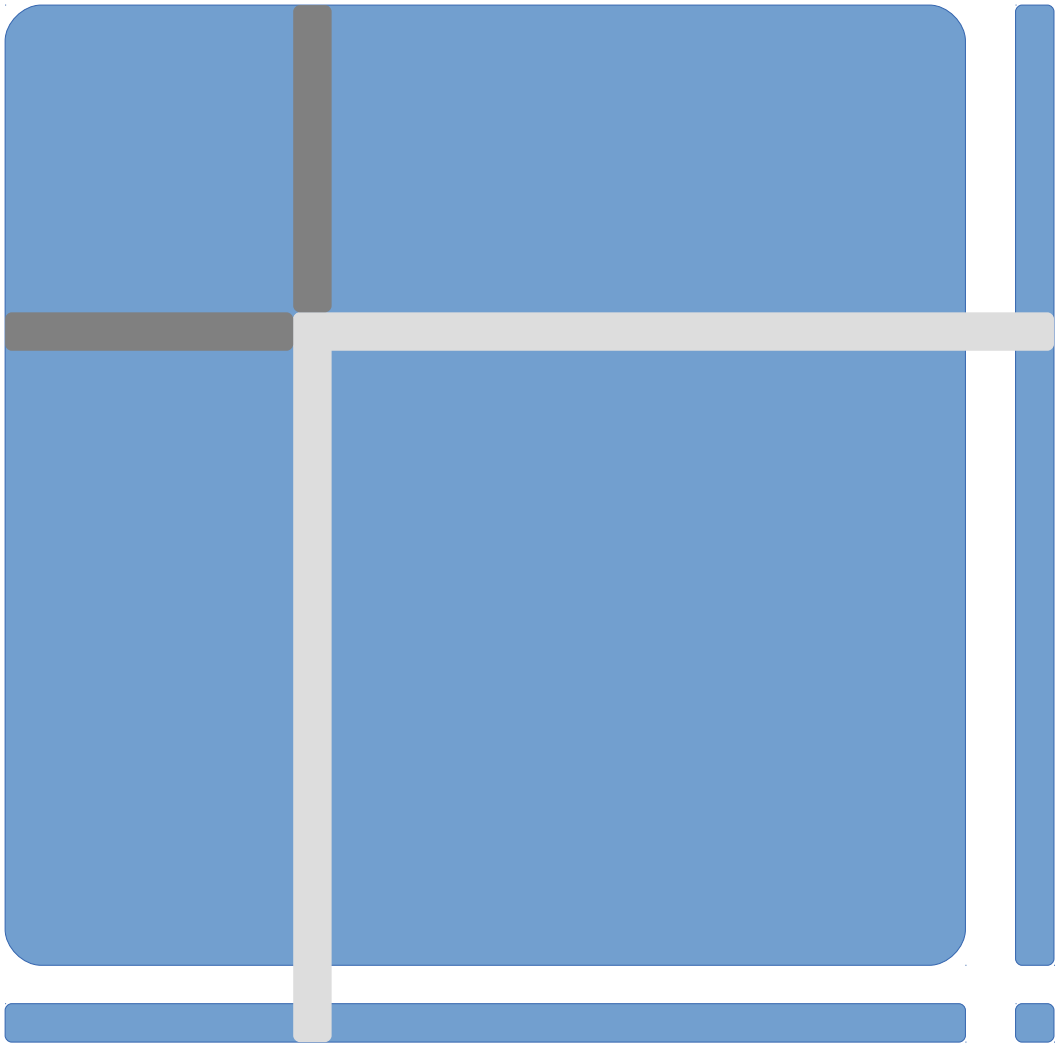


Trailing update

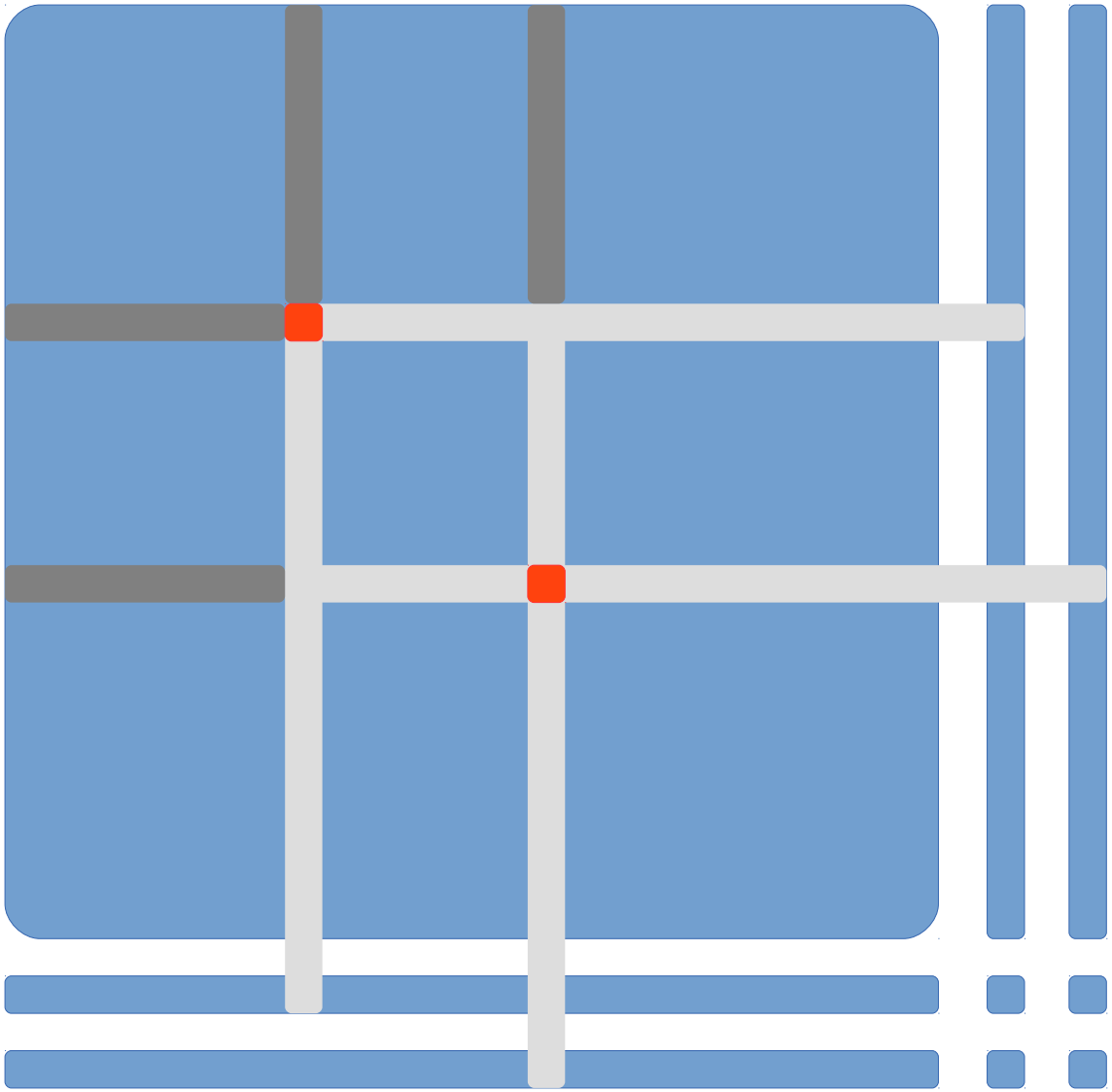


End iteration

HRD: Extra Computation for Single-Error Protection

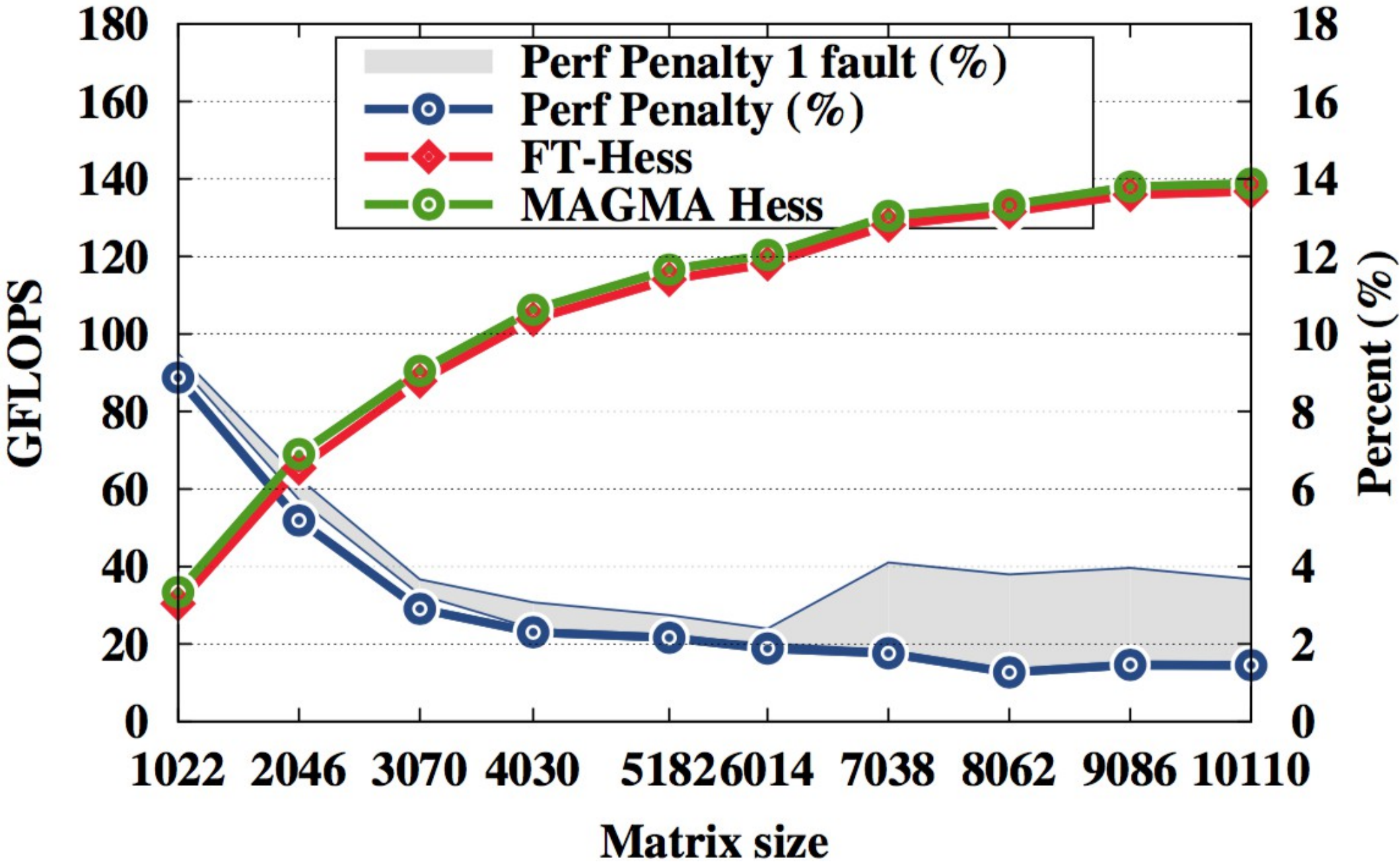


HRD: Extra Computation for Two-Error Protection



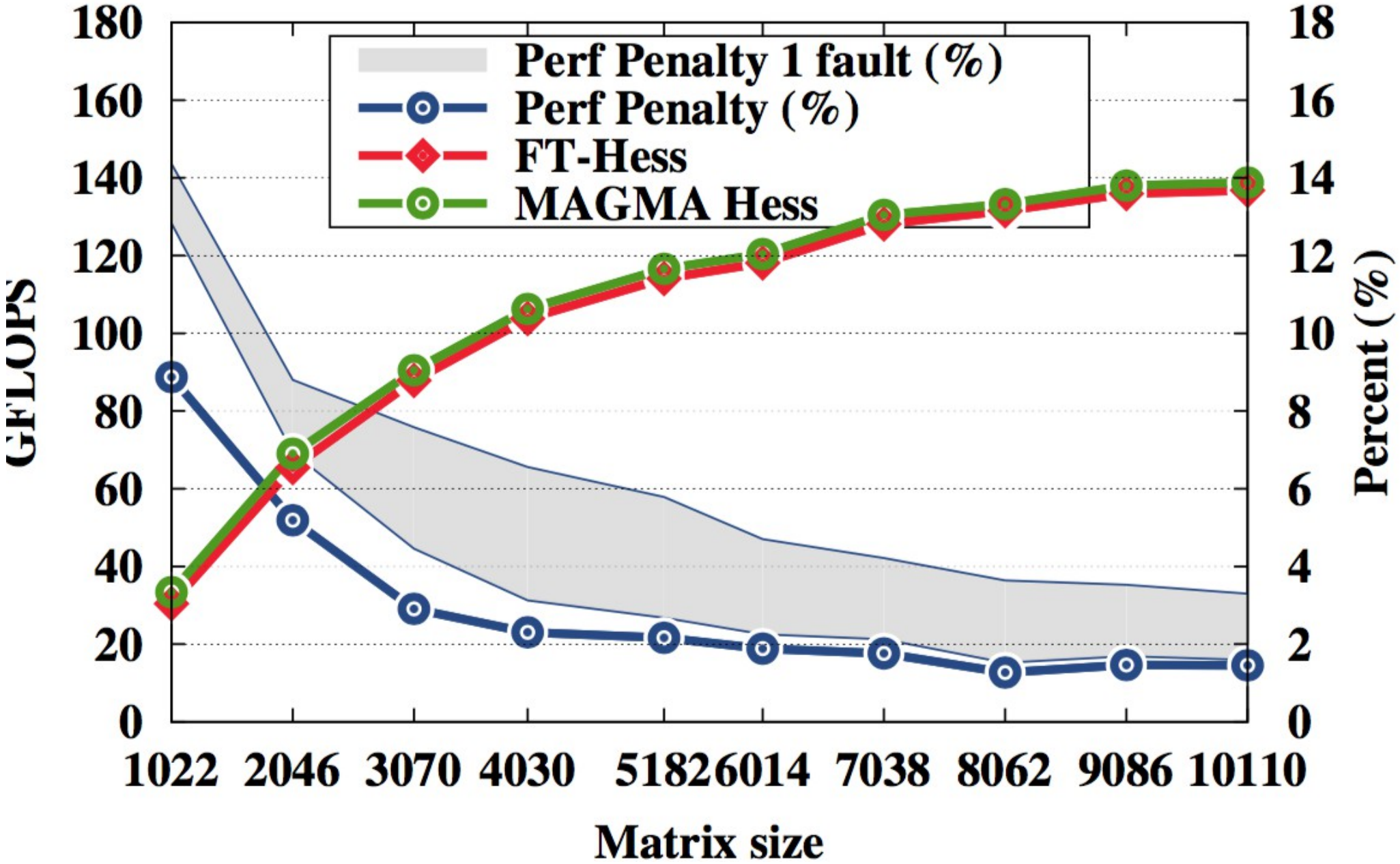
DGETRF (solve)

Performance with Error Protection (Error in Panel)



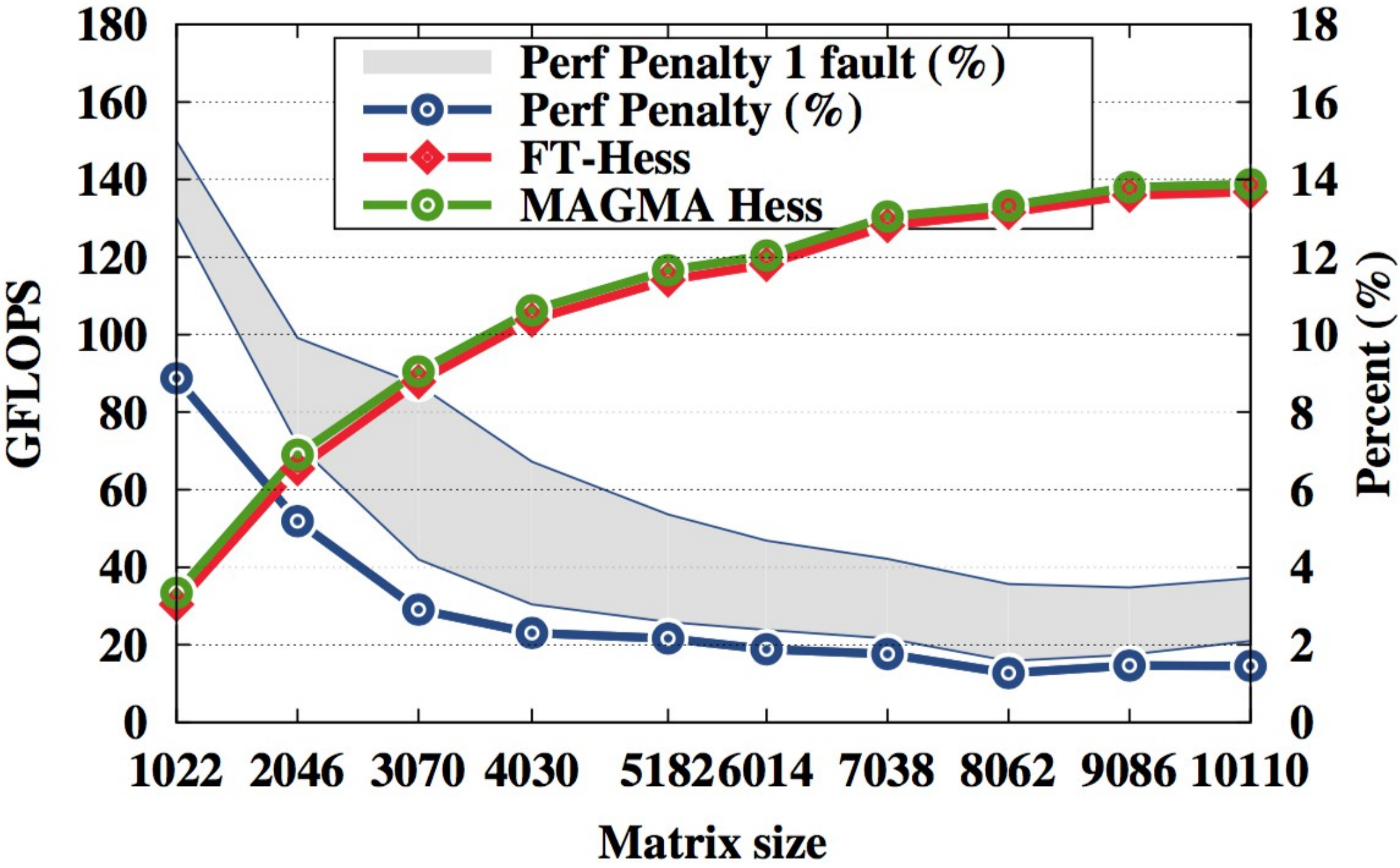
Intel Sandy Bridge Xeon E5-2670 2.6 GHz+MKL 11.2 NVIDIA Kepler K40c+cuBLAS 7

Performance with Error Protection (Error in Upper)



Intel Sandy Bridge Xeon E5-2670 2.6 GHz+MKL 11.2 NVIDIA Kepler K40c+cuBLAS 7

Performance with Error Protection (Error in Trailing)



Intel Sandy Bridge Xeon E5-2670 2.6 GHz+MKL 11.2 NVIDIA Kepler K40c+cuBLAS 7

Numerical Accuracy

- Numerical accuracy can be measured in various forms
  - Scaled residual (backward error)
  - Orthogonality of Q's
- Accuracy can be different depending on:
  - Location of error: panel, upper, trailing
  - Time of error: beginning, middle, end of HRD
- Summary of numerical results for N=1k,…,10k
  - Errors in non-fault tolerant code: 10^{-18} – 10^{-17}
  - Errors in upper and trailing on the order of 10^{-18} – 10^{-17}
  - Errors in panel on the order of 10^{-15} – 10^{-14}

Conclusions and Future Work

- Summary
  - Presented design and analysis of fault-tolerant Hessenberg reduction
  - The methods used: ABFT, diskless checkpointing
  - Hardware used: GPU accelerator
  - Minimal overhead in performance
  - About 2-digit loss for some scenarios but still accurate in working precision
- Future directions
  - Address all two-sided factorizations within a single framework
  - Support for upcoming accelerators:
    - Intel KNL and Sky Lake
    - NVIDIA Pascal, Tegra, Jetson, Denver2
    - AMD Polaris, Zen
    - Google TPU