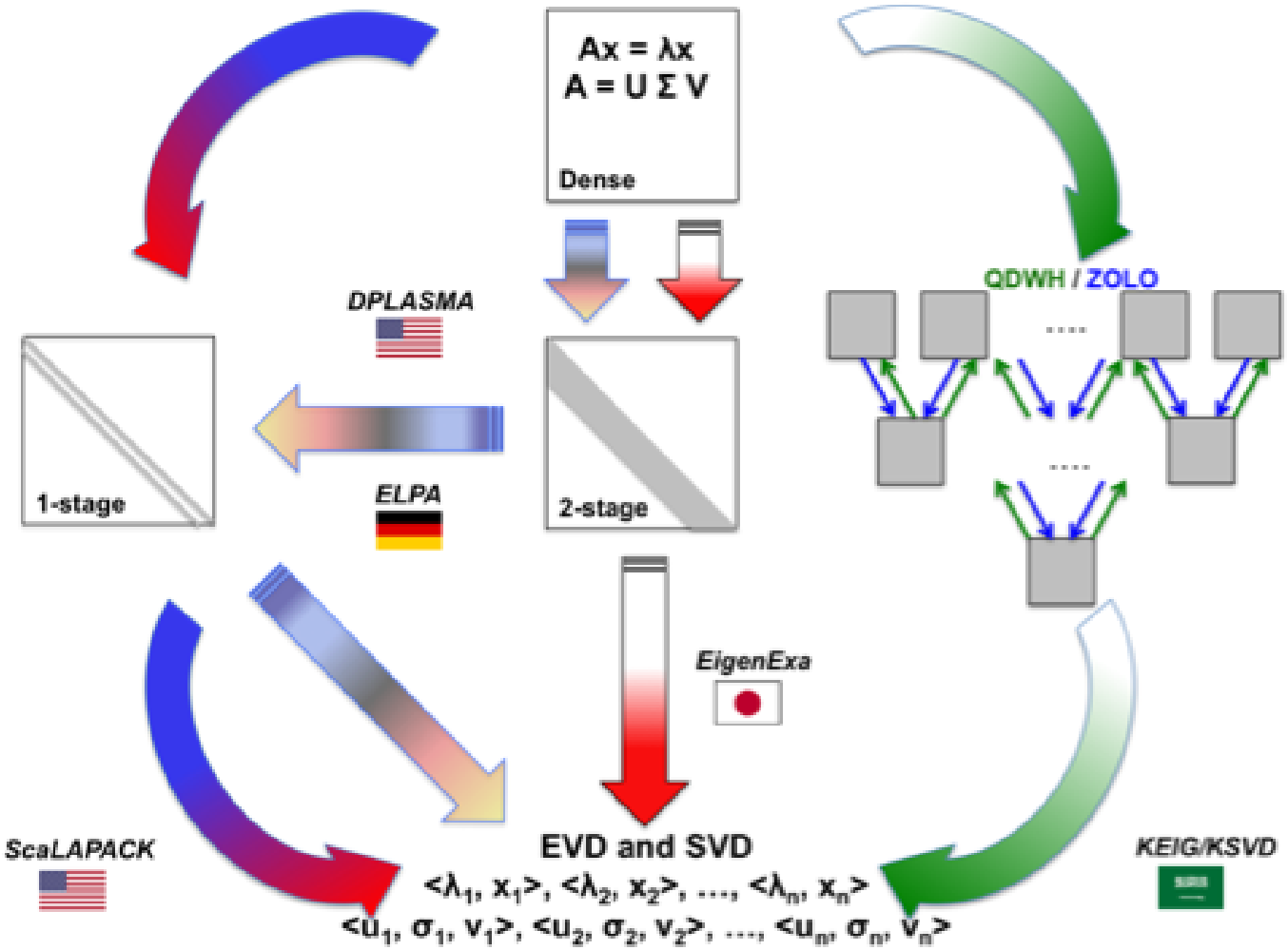
High Performance Polar Decomposition on Manycore Systems and its application to Symmetric Eigensolvers and the Singular Value Decomposition

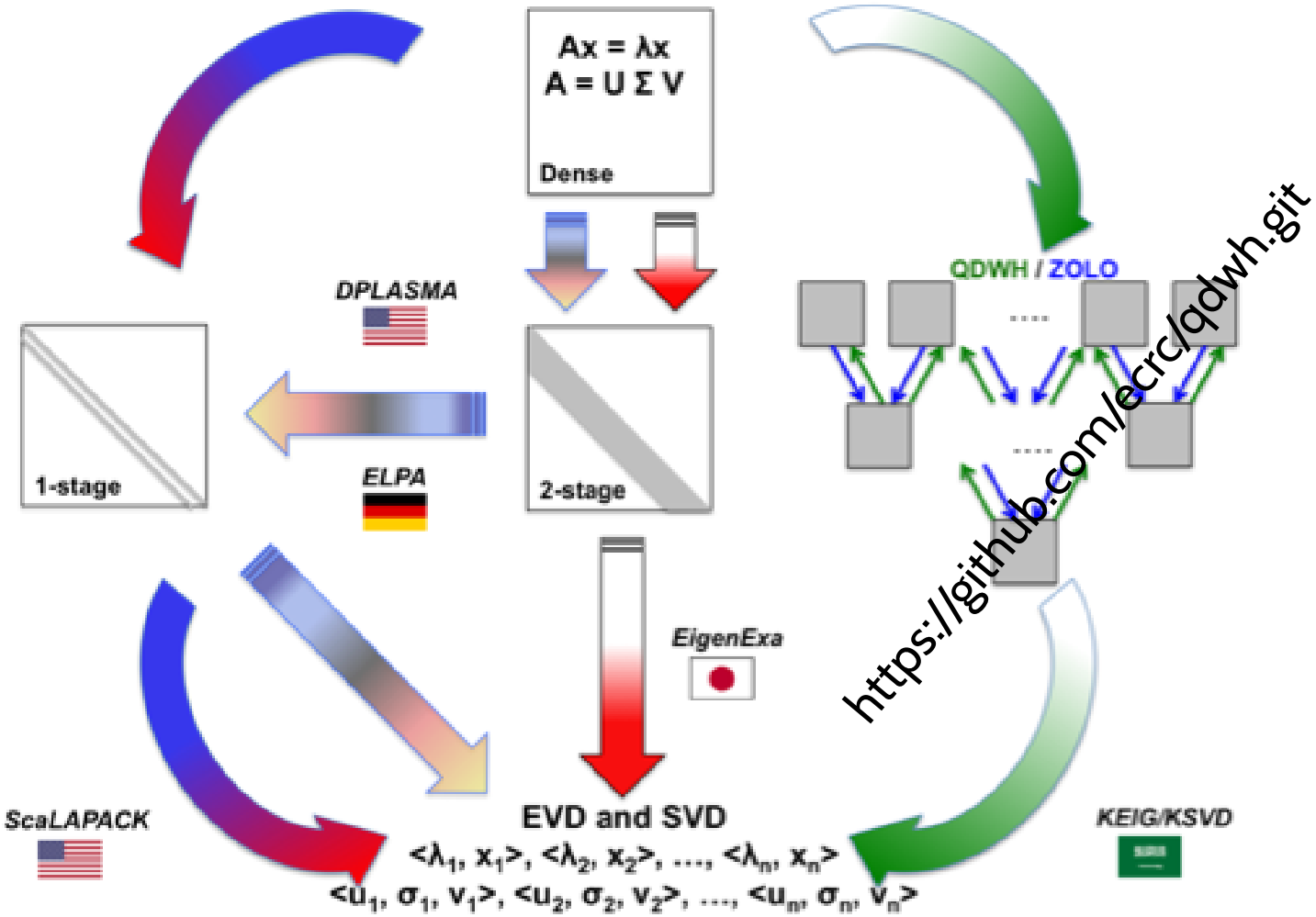
D. Sukkari[1], H. Ltaief[2] and D. Keyes[2]
[1] Innovative Computing Laboratory @ UTK, USA
[2]Extreme Computing Research Center @ KAUST, Saudi Arabia

SIAM CSE'21
March 1 - March 5, 2021

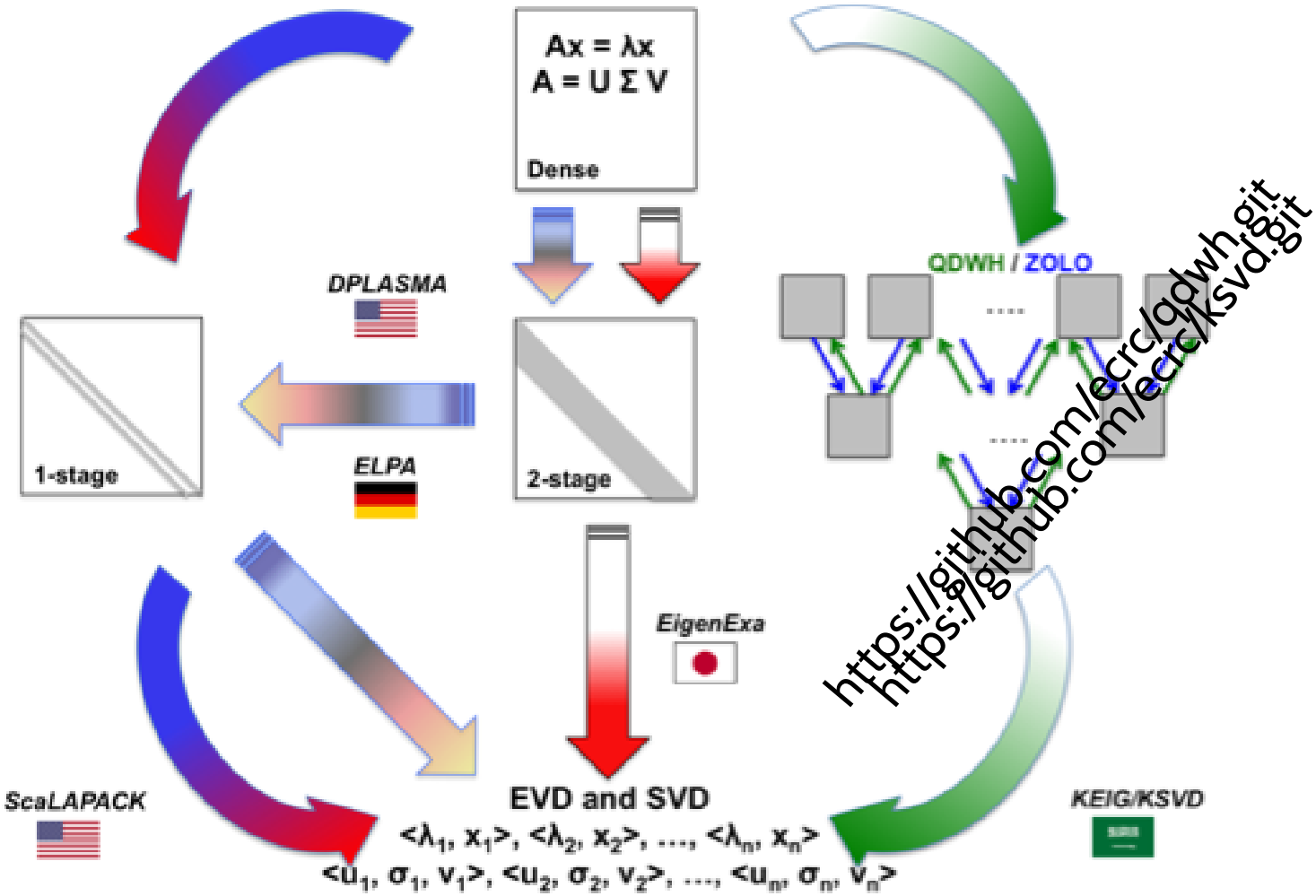
The Big Picture



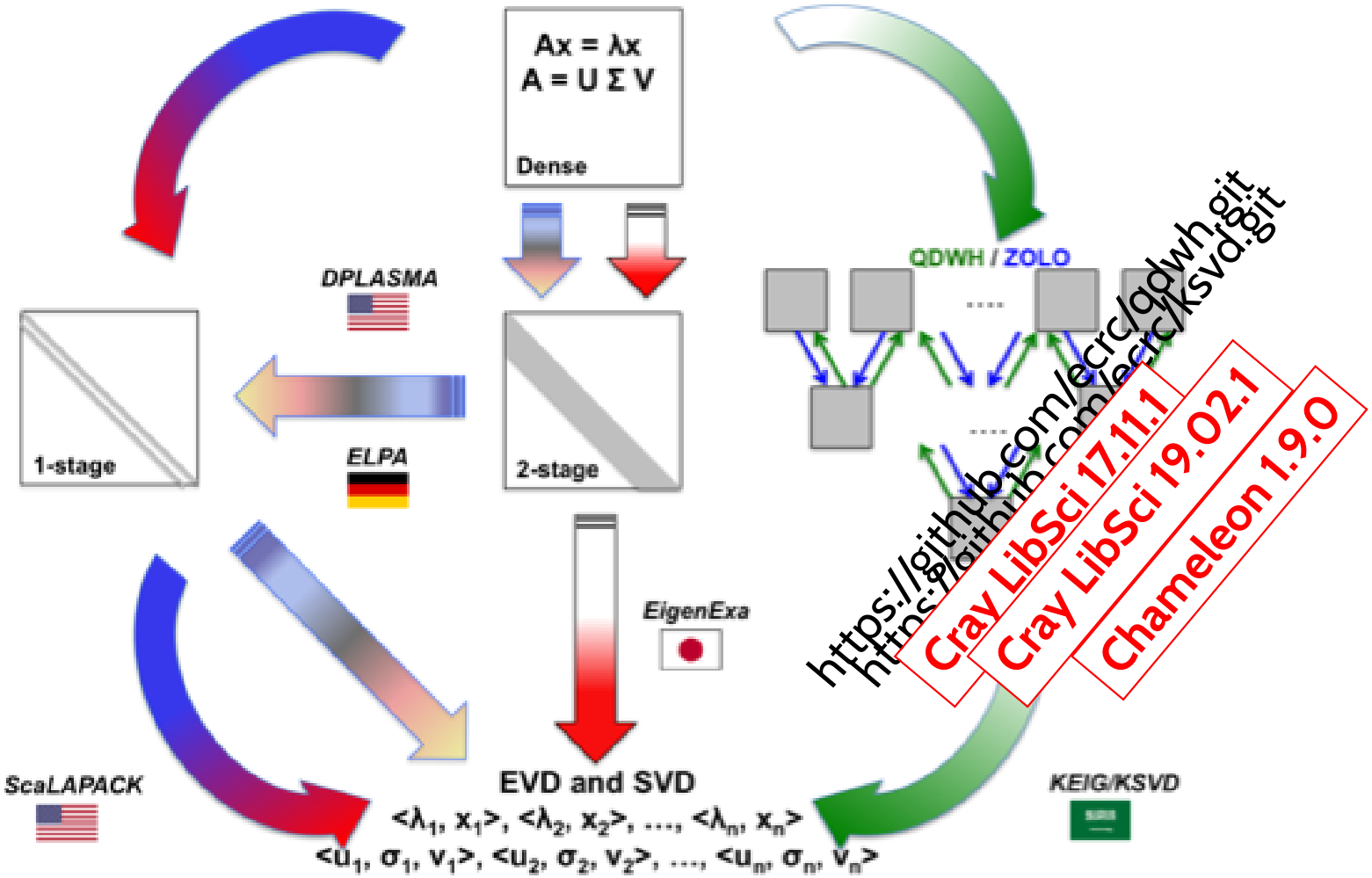
The Big Picture



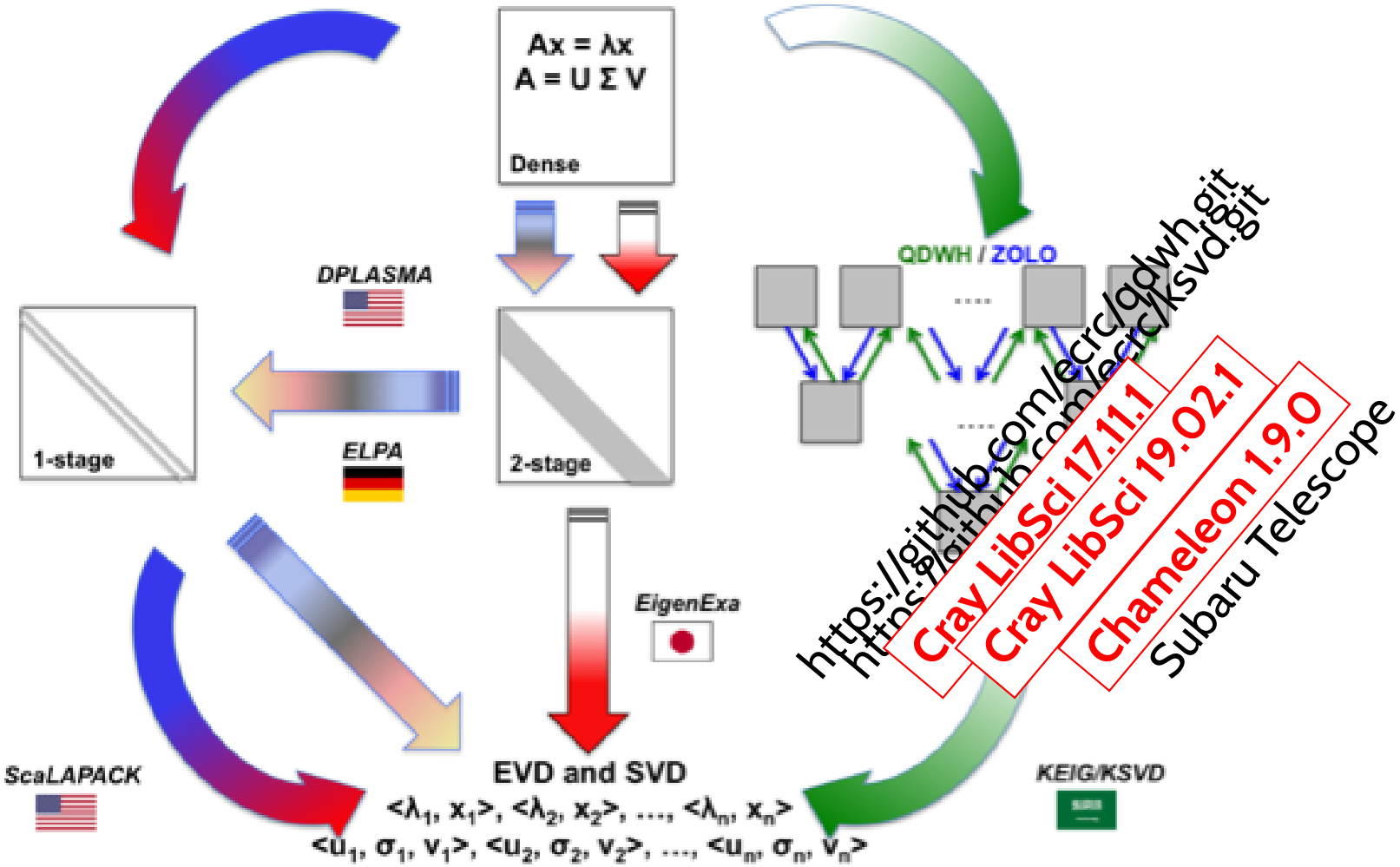
The Big Picture



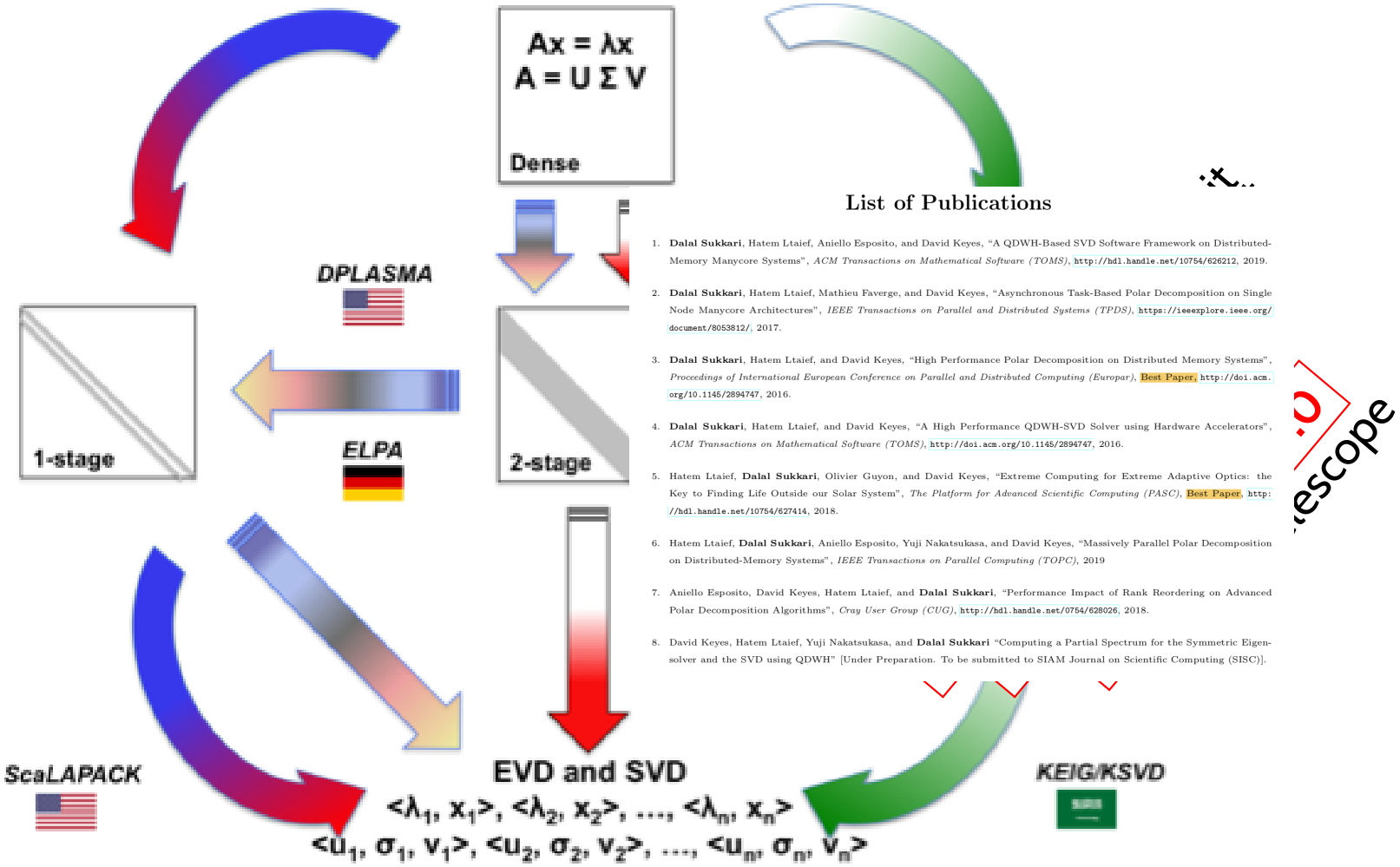
The Big Picture



The Big Picture



The Big Picture



List of Publications

1. **Dalal Sukkari**, Hatem Ltaief, Aniello Esposito, and David Keyes, "A QDWH-Based SVD Software Framework on Distributed-Memory Manycore Systems", *ACM Transactions on Mathematical Software (TOMS)*, http://hdl.handle.net/10754/626212, 2019.

2. **Dalal Sukkari**, Hatem Ltaief, Mathieu Faverge, and David Keyes, "Asynchronous Task-Based Polar Decomposition on Single Node Manycore Architectures", *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, https://ieeexplore.ieee.org/document/8053812/, 2017.

3. **Dalal Sukkari**, Hatem Ltaief, and David Keyes, "High Performance Polar Decomposition on Distributed Memory Systems", *Proceedings of International European Conference on Parallel and Distributed Computing (Europar)*, Best Paper, http://doi.acm.org/10.1145/2894747, 2016.

4. **Dalal Sukkari**, Hatem Ltaief, and David Keyes, "A High Performance QDWH-SVD Solver using Hardware Accelerators", *ACM Transactions on Mathematical Software (TOMS)*, http://doi.acm.org/10.1145/2894747, 2016.

5. Hatem Ltaief, **Dalal Sukkari**, Olivier Guyon, and David Keyes, "Extreme Computing for Extreme Adaptive Optics: the Key to Finding Life Outside our Solar System", *The Platform for Advanced Scientific Computing (PASC)*, Best Paper, http://hdl.handle.net/10754/627414, 2018.

6. Hatem Ltaief, **Dalal Sukkari**, Aniello Esposito, Yuji Nakatsukasa, and David Keyes, "Massively Parallel Polar Decomposition on Distributed-Memory Systems", *IEEE Transactions on Parallel Computing (TOPC)*, 2019

7. Aniello Esposito, David Keyes, Hatem Ltaief, and **Dalal Sukkari**, "Performance Impact of Rank Reordering on Advanced Polar Decomposition Algorithms", *Cray User Group (CUG)*, http://hdl.handle.net/0754/628026, 2018.

8. David Keyes, Hatem Ltaief, Yuji Nakatsukasa, and **Dalal Sukkari** "Computing a Partial Spectrum for the Symmetric Eigensolver and the SVD using QDWH" [Under Preparation. To be submitted to SIAM Journal on Scientific Computing (SISC)].

Outline

1. Motivation
2. State-of-the-Art Approach
3. QDWH-based Polar Decomposition
4. Algorithmic Designs of QDWH
5. Application to Full/Partial SVD
6. Application to Partial EIG
7. Application to Extreme Adaptive Optics
8. Conclusion and Future Work

Outline

1. Motivation
2. State-of-the-Art Approach
3. QDWH-based Polar Decomposition
4. Algorithmic Designs of QDWH
5. Application to Full/Partial SVD
6. Application to Partial EIG
7. Application to Extreme Adaptive Optics
8. Conclusion and Future Work

Problem Statement

The polar decomposition:

$$\mathrm{A} = \mathrm{U_pH}\,,\ A \in \mathbb{R}^{m\times n}(m \geq n),$$

where U_p is an orthogonal matrix and $H = \sqrt{A^\top A}$ is a symmetric positive semidefinite matrix

- Using the singular value decomposition (SVD):

$$A = U\Sigma V^\top = (UV^\top)(V\Sigma V^\top) = U_pH$$

- Using the inverse free QR dynamically-weighted Halley iteration (QDWH) or ZOLO-PD (Zolotarev, 1877)

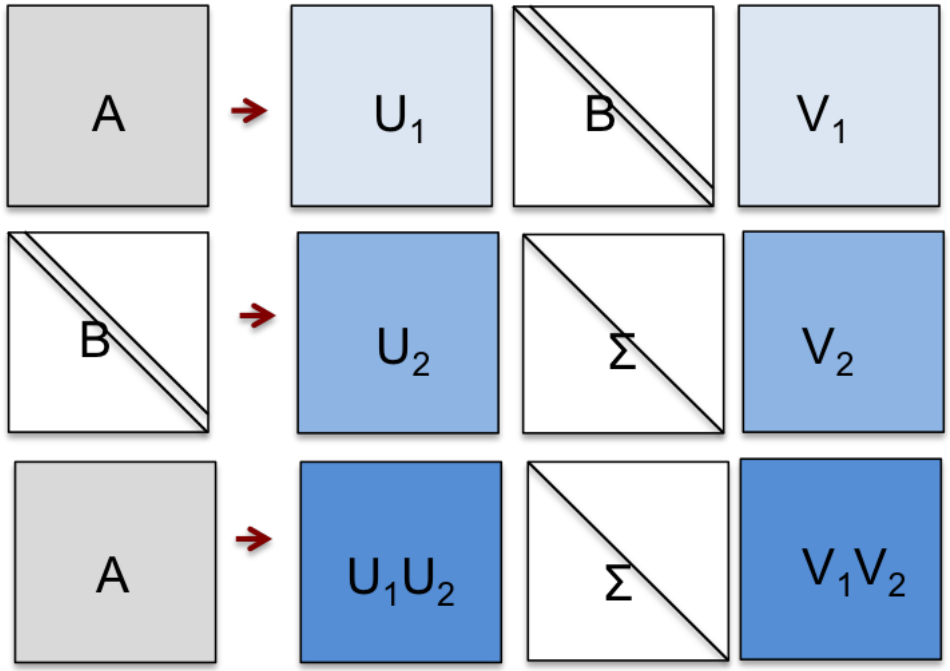
Why the Polar Decomposition is Needed?

- Represents a critical numerical algorithm for various applications, including continuum mechanics, aerospace computations, chemistry
- Stands as a building block for the dense symmetric eigensolver and SVD
- Leverages the computations of partial eigenspectrum and most significant singular values/vectors

Outline

1. Motivation
2. **State-of-the-Art Approach**
3. QDWH-based Polar Decomposition
4. Algorithmic Designs of QDWH
5. Application to Full/Partial SVD
6. Application to Partial EIG
7. Application to Extreme Adaptive Optics
8. Conclusion and Future Work

Standard 1-stage Approach (LAPACK)



This form the standard SVD solver based on the QR algorithm or the divide-and-conquer method.

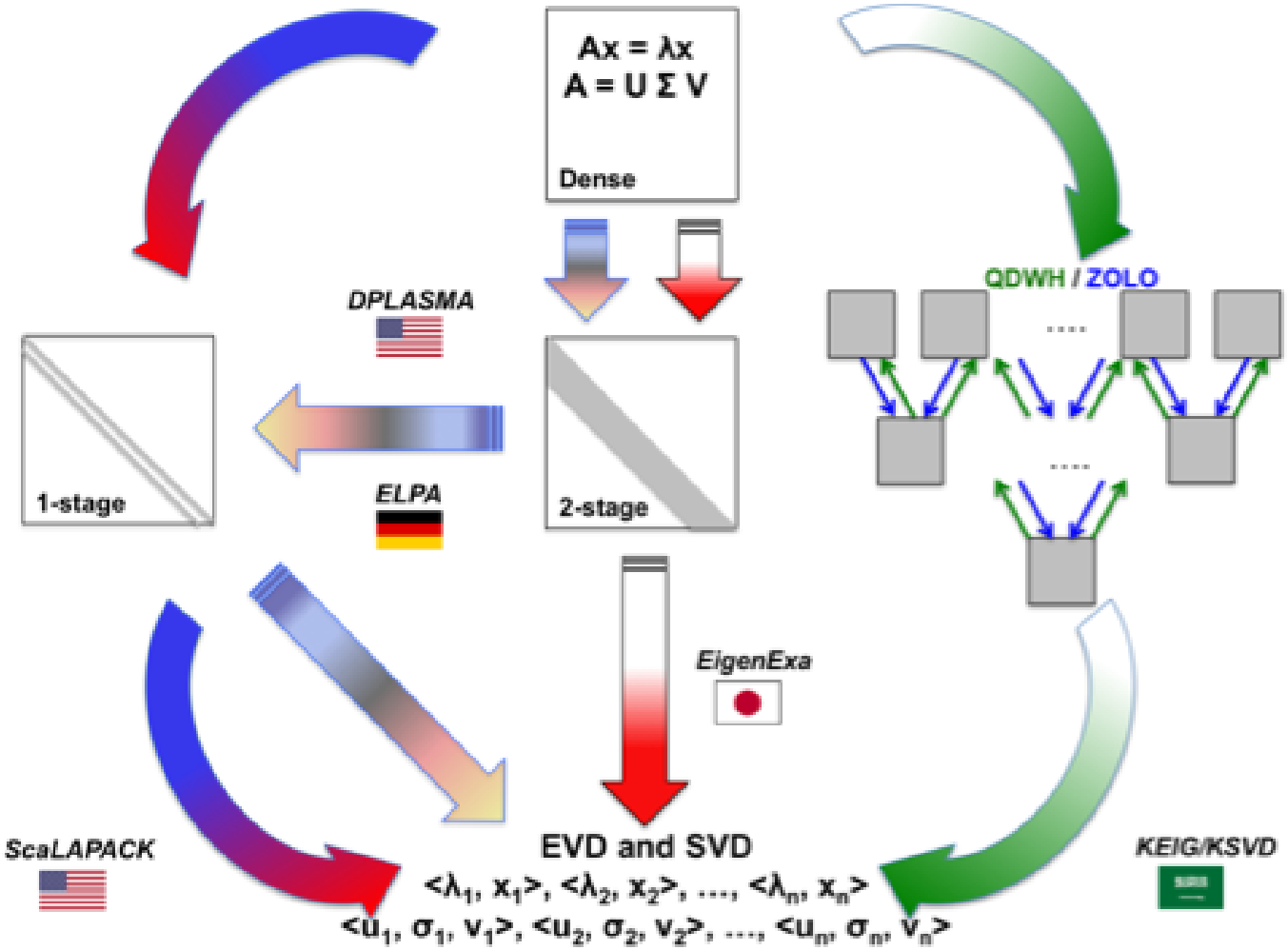
	Bidiagonal Reduction $(8/3n^3)$	DGESVD $(\Sigma)(8/3n^3)$	DGESVD $(U\Sigma V^\top)(17n^3)$
Level-2 BLAS $(4/3n^3)$	$50\% flops$ $90\% time$	$50\% flops$ $85\% time$	$8\% flops$ $30\% time$

Hardware Trends: Energy Matters!

	2011	2018
DP FLOP	100 pJ	10 pJ
DP DRAM Read	4800 pJ	1920 pJ
Local interconnect	7500 pJ	2500 pJ
Cross system	9000 pJ	3500 pJ

John Shalf, LBNL

The Big Picture



Outline

1. Motivation
2. State-of-the-Art Approach
3. **QDWH-based Polar Decomposition**
4. Algorithmic Designs of QDWH
5. Application to Full/Partial SVD
6. Application to Partial EIG
7. Application to Extreme Adaptive Optics
8. Conclusion and Future Work

QDWH-based Polar Decomposition

Algorithm 1 QDWH [1]

1: Estimate **the condition number** l_0 of the matrix A using **LU** factorization
2: Scale the matrix $X_0 = A/\alpha$, $\alpha \geqslant \|A\|_2$
3: **while** $\|X_k - X_{k+1}\| \geq tol$ **do**
4: Compute the scalars l_k, a_k, b_k, c_k
5: **if** $c_k > 100$ **then**

$$\begin{bmatrix} \sqrt{c_k}X_k \\ I \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R,\ X_{k+1} = \frac{b_k}{c_k}X_k + \frac{1}{\sqrt{c_k}}(a_k - \frac{b_k}{c_k})Q_1 Q_2^\top$$

6: **else**

$$X_{k+1} = \frac{b_k}{c_k}X_k + \left(a_k - \frac{b_k}{c_k}\right)(X_k W_k^{-1})W_k^{-\top},\ W_k = \mathsf{chol}(Z_k),\ Z_k = I + c_k X_k^\top X_k$$

7: $U_p = X_{k+1}$
8: Compute $H = U_p^\top A$

[1] Yuji Nakatsukasa, Nicholas J. Higham: Stable and Efficient Spectral Divide and Conquer Algorithms for the Symmetric Eigenvalue Decomposition and the SVD. SIAM J. Scientific Computing 35(3) (2013)

QDWH Algorithmic Complexity

▶ The total flop count for QDWH depends on the condition number of the matrix κ:

$$\#flops = \frac{8}{3}n^3 + (8 + \frac{2}{3})n^3 \times \#it_{QR} + \quad (4 + \frac{1}{3})n^3 \times \#it_{Chol} + 2n^3, \tag{1}$$

where $\#it_{QR}$ and $\#it_{Chol}$ correspond to the number of QR-based and Cholesky-based iterations, respectively.

κ	1	10^{16}
flops	$(13 + \frac{1}{3})n^3$	$(43 + \frac{2}{3})n^3$

Historical Timeline

1877 Zolotarev, best rational approximant for the scalar sign function

1902 Autonne, *SMF, definition of the polar decomposition*

1994 Higham and Papadimitriou, *SIAM, matrix inversion QDWH, shared-memory systems*

2010 Nakatsukasa et. al, *SIAM, inverse-free QDWH, theoretical accuracy study*

2013 Nakatsukasa and Higham, *SIAM, QDWH-EIG, QDWH-SVD, theoretical accuracy study*

2014 Nakatsukasa, *SIAM, ZOLO-PD, ZOLO-SVD, ZOLO-EIG, theoretical accuracy study*

2016 Sukkari, Ltaief and Keyes, *TOMS, QDWH-SVD, block algorithm, shared-memory system equipped with multiple GPUs*

2016 Sukkari, Ltaief and Keyes, *Euro-Par and TOMS, QDWH, QDWH-SVD, block algorithm, distributed-memory system*

2017 Sukkari, Ltaief, Faverge and Keyes, *TPDS, QDWH, task-based, shared-memory system equipped with multiple GPUs*

2018 Ltaief, Sukkari, Guyon and Keyes, *PASC, QDWH-SVD, block algorithm, shared-memory system equipped with multiple GPUs*

2019 Ltaief, Sukkari, Esposito, Nakatsukasa and Keyes, *TOPC, ZOLO-PD, block algorithm, distributed-memory system*

But Here is the Good News!

- Performs *extra* flops but *nice* flops
- Relies on compute intensive kernels
- Exposes high concurrency
- Maps well to current hardware architectures (e.g., GPUs)
- Minimizes data movement (i.e., communication-avoidance)
- Weakens resource synchronizations (i.e., synchronization-avoidance)

Historical Timeline

1877 Zolotarev, best rational approximant for the scalar sign function

1902 Autonne, *SMF, definition of the polar decomposition*

1994 Higham and Papadimitriou, *SIAM, matrix inversion QDWH, shared-memory systems*

2010 Nakatsukasa et. al, *SIAM, inverse-free QDWH, theoretical accuracy study*

2013 Nakatsukasa and Higham, *SIAM, QDWH-EIG, QDWH-SVD, theoretical accuracy study*

2014 Nakatsukasa, *SIAM, ZOLO-PD, ZOLO-SVD, ZOLO-EIG, theoretical accuracy study*

2016 Sukkari, Ltaief and Keyes, *TOMS, QDWH-SVD, block algorithm, shared-memory system equipped with multiple GPUs*

2016 Sukkari, Ltaief and Keyes, *Euro-Par and TOMS, QDWH, QDWH-SVD, block algorithm, distributed-memory system*

2017 Sukkari, Ltaief, Faverge and Keyes, *TPDS, QDWH, task-based, shared-memory system equipped with multiple GPUs*

2018 Ltaief, Sukkari, Guyon and Keyes, *PASC, QDWH-SVD, block algorithm, shared-memory system equipped with multiple GPUs*

2019 Ltaief, Sukkari, Esposito, Nakatsukasa and Keyes, *TOPC, ZOLO-PD, block algorithm, distributed-memory system*

Outline

1. Motivation
2. State-of-the-Art Approach
3. QDWH-based Polar Decomposition
4. **Algorithmic Designs of QDWH**
5. Application to Full/Partial SVD
6. Application to Partial EIG
7. Application to Extreme Adaptive Optics
8. Conclusion and Future Work

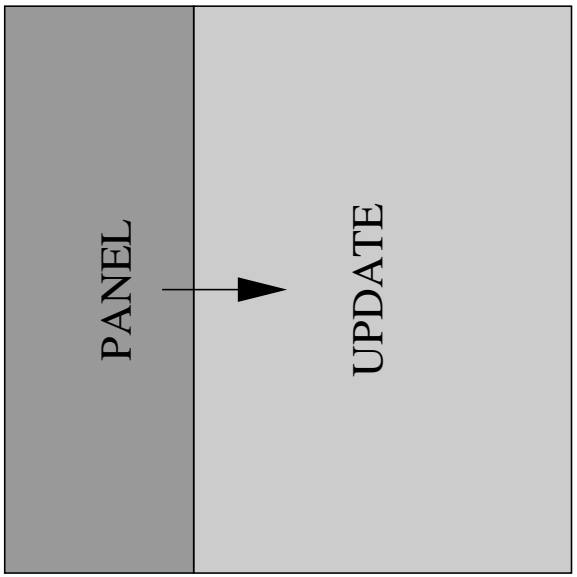
ScaLAPACK 101

- The distributed-memory implementation of QDWH using **ScaLAPACK**:
  - Relies on block algorithms, similar to LAPACK
  - Employs the bulk synchronous programming model
  - Uses 2D block cyclic data distribution to map the matrix data to the distributed-memory
  - Relies on the Message Passing Interface (MPI) to exchange data within the grid of MPI processes

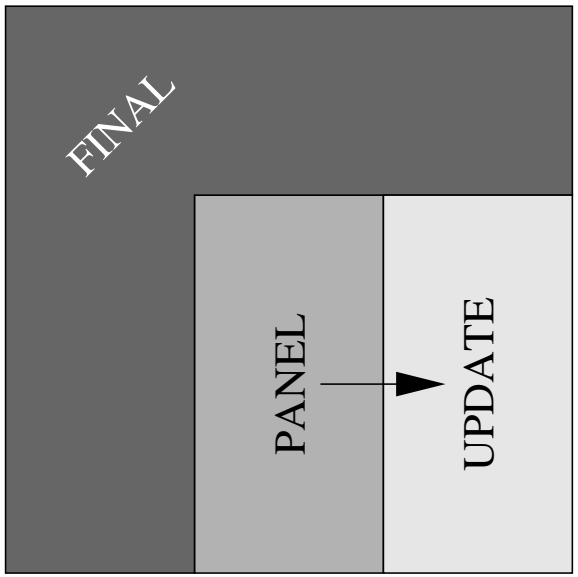
ScaLAPACK 101

ScaLAPACK Block Algorithm:

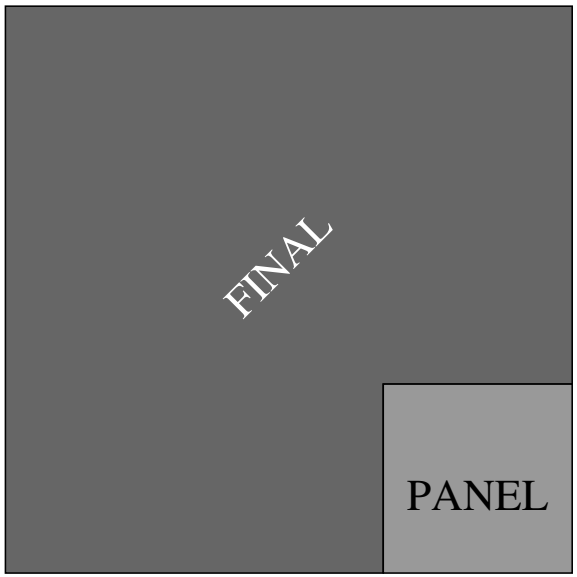
- Panel-Update Sequence
- Transformations are blocked/accumulated within the Panel (Level 2 PBLAS)
- Transformations applied at once on the trailing submatrix (Level 3 PBLAS)
- Parallelism hidden inside the PBLAS



(a) First step.



(b) Second step.



(c) Third step.

Environment Settings

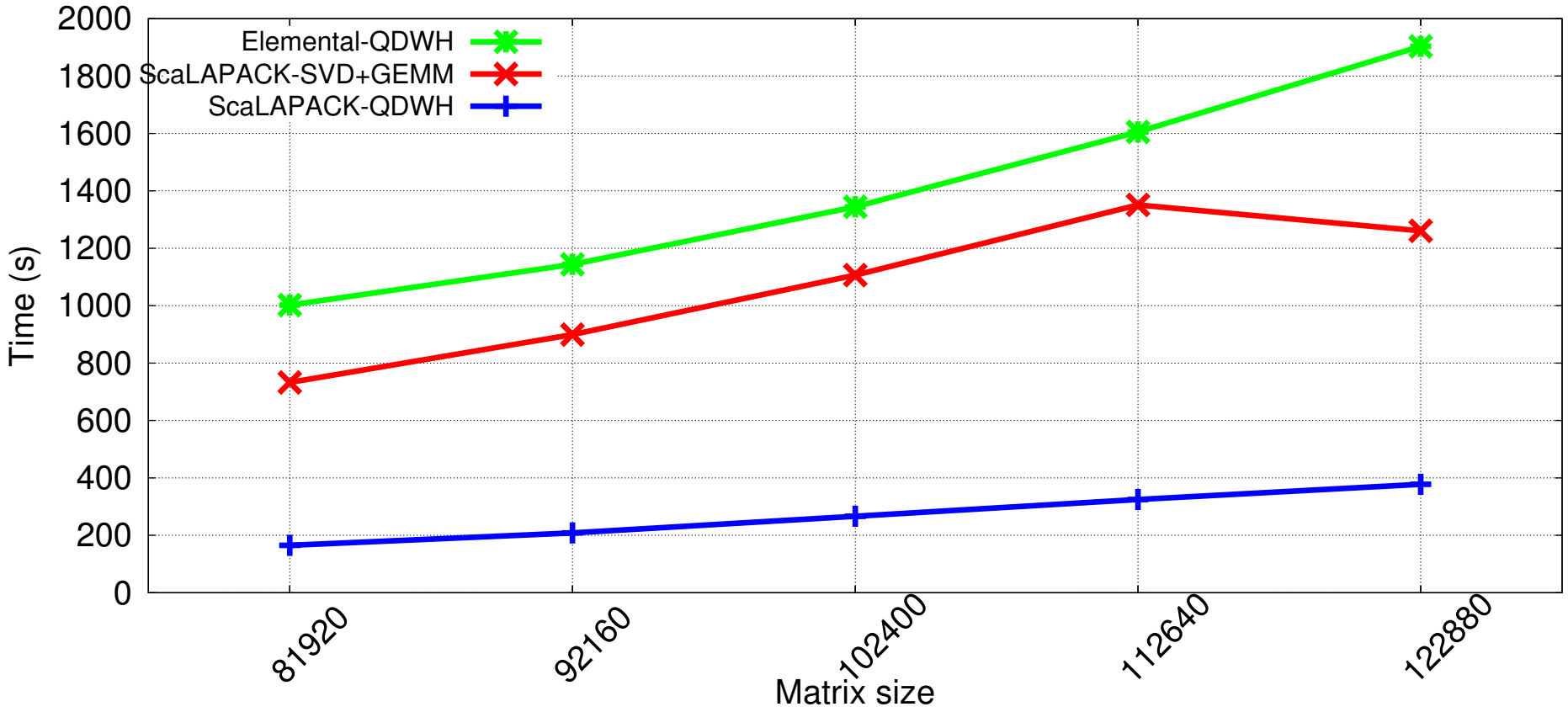
Software:

- Intel Compiler Suites v$15.0.2.164$ and v$17.0.4.196$ on *Shaheen-2*.
- Cray LibSci/13.2.0 ScaLAPACK library
- MPICH library
- Ill Conditioned matrices generated using *pdmatgen* ScaLAPACK routine with a condition number $\kappa = 10^{16}$.

Hardware: The Cray XC40 system codenamed *Shaheen-2* installed at the KAUST Supercomputing Laboratory (KSL). *Shaheen-2* has 6174 compute nodes, each node has:

- Two-sockets 16-core Intel Xeon E5-2690 v3 (Haswell)
- The base frequency is at 2.3GHz
- 128GB of DDR3 main memory

Performance Comparison 128x288 (proc grid)



[3] Dalal Sukkari, Hatem Ltaief, Aniello Esposito, and David Keyes: A QDWH-Based SVD Software Framework on Distributed-Memory Manycore Systems. ACM Transactions on Mathematical Software (TOMS), (2019)

CRAY THE SUPERCOMPUTER COMPANY

Outline

1. Motivation
2. State-of-the-Art Approach
3. QDWH-based Polar Decomposition
4. Algorithmic Designs of QDWH
5. **Application to Full/Partial SVD**
6. Application to Partial EIG
7. Application to Extreme Adaptive Optics
8. Conclusion and Future Work

QDWH-based SVD

Algorithm 2 QDWH-SVD [1]

1: Compute **the polar decomposition** $A = U_p H$ via QDWH
2: Compute the symmetric eigendecomposition $H = V \Sigma V^\top$
3: Form $U = U_p V$

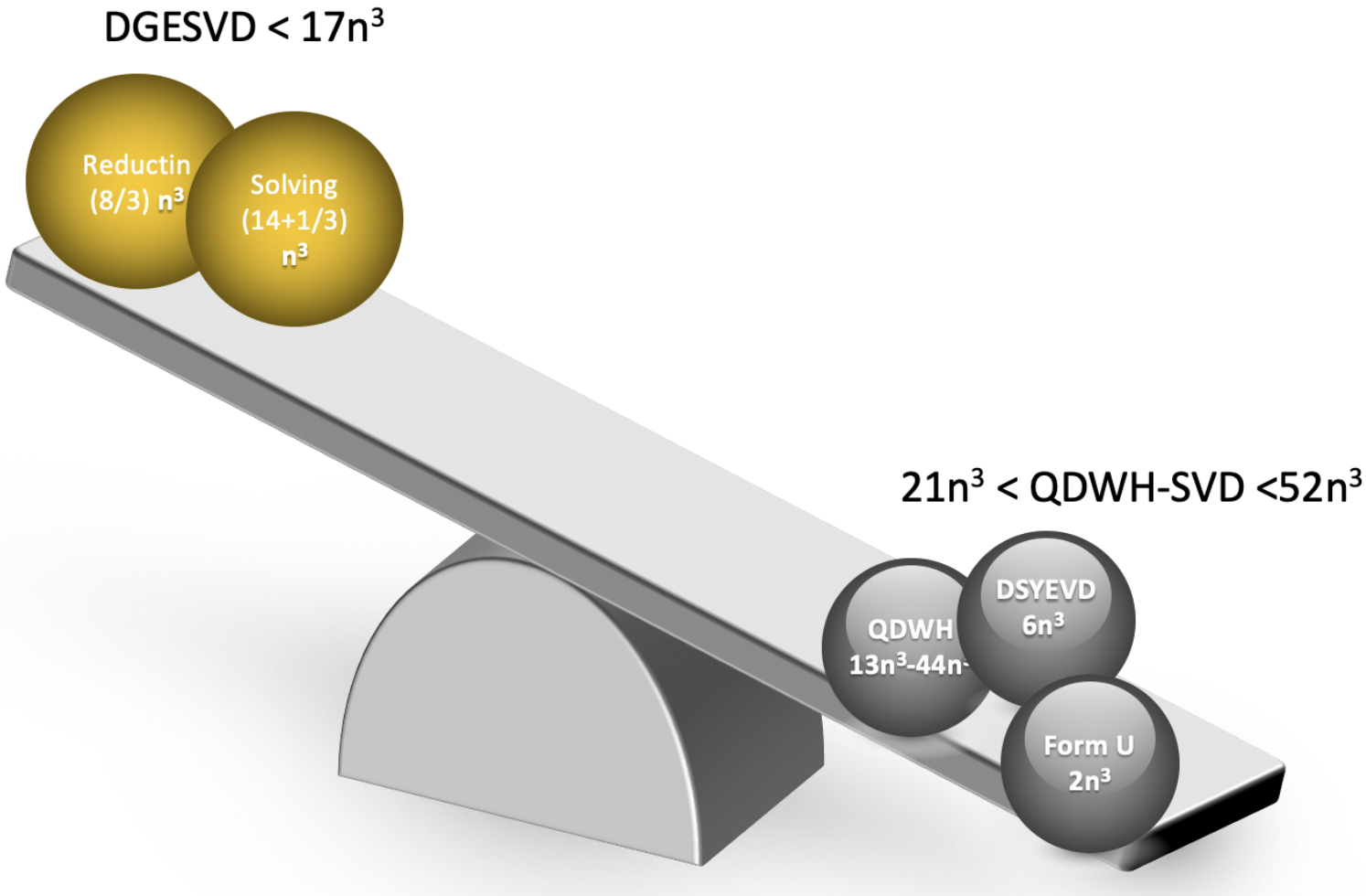
[1] Yuji Nakatsukasa, Nicholas J. Higham: Stable and Efficient Spectral Divide and Conquer Algorithms for the Symmetric Eigenvalue Decomposition and the SVD. SIAM J. Scientific Computing 35(3) (2013)

QDWH-SVD

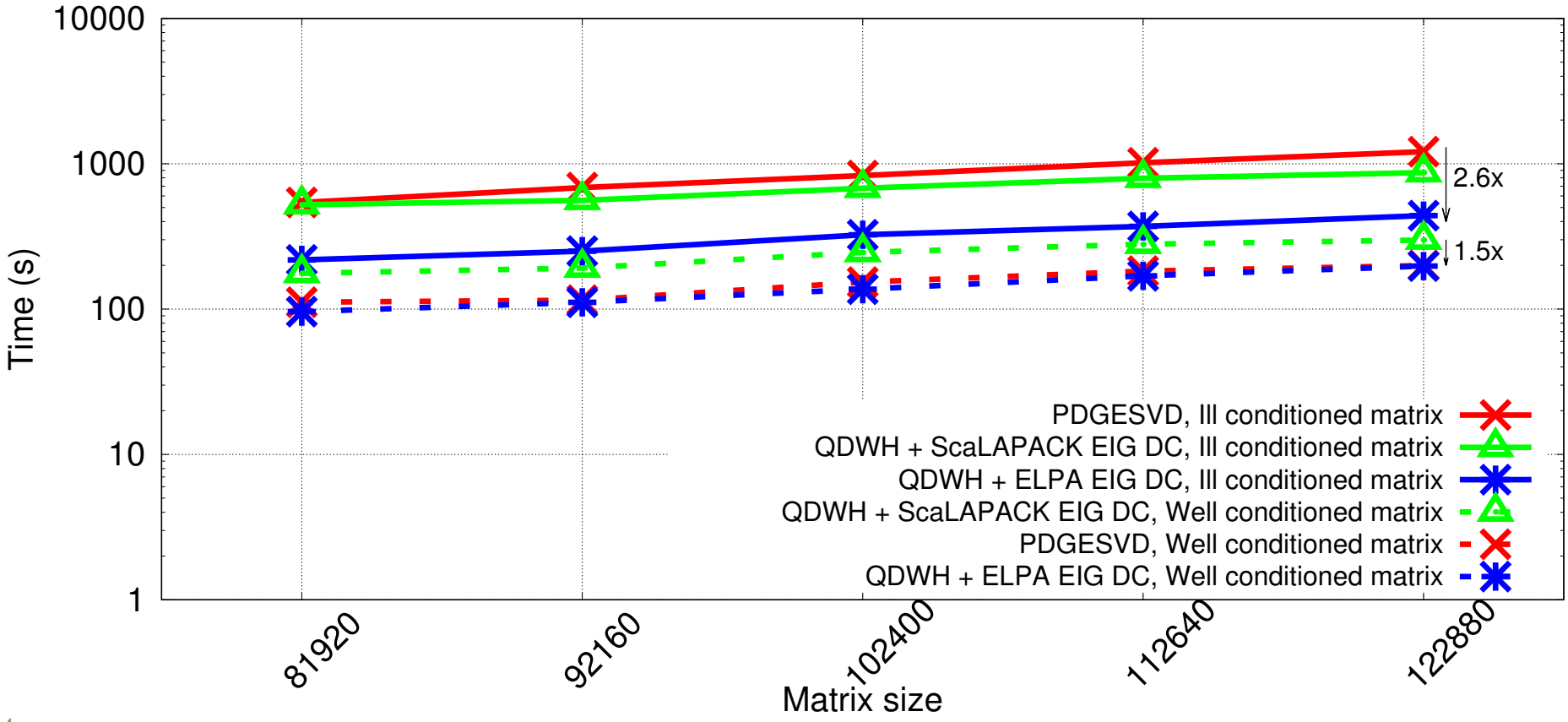
The eigensolvers to build ScaLAPACK-QDWH-SVD:

- PDSYEVD ScaLAPACK eigensolver based on a 1-stage tridiagonal reduction and the DC eigensolver
- ELPA-EIG which combines a two-stage reduction with a divide-and-conquer eigensolver

QDWH-SVD Algorithmic Complexity



Performance Comparison 128x288 (proc grid)



[4] Dalal Sukkari, Hatem Ltaief, and David Keyes: High Performance Polar Decomposition on Distributed Memory Systems. Proceedings of International European Conference on Parallel and Distributed Computing (EuroPar), Best Paper, 2016

CRAY THE SUPERCOMPUTER COMPANY

Extension to Partial SVD

$$A = U_{1:s} \Sigma_{1:s} {V_{1:s}}^{\top}$$

$$A \in \mathbb{R}^{m \times n} (m \geq n)$$

where:
s is the number of the most significant singular values, Σ is the matrix of the singular values, and
U, V are the matrices of the corresponding singular vectors.

QDWHpartial-SVD

Algorithm 3 The QDWH-Based Partial SVD.

1: Compute **the polar decomposition** $A = U_p H$ using **QDWH**
2: Calculate $[Q\ R] = QR(Id - U_p U_p)$
3: Find the index $ind = min(find(abs(diag(R)) < threshold(s)))$
4: Extract $\tilde{Q} = Q(:, ind : end)$
5: Reduce the original matrix problem $\tilde{A} = A\tilde{Q}$
6: Compute the SVD of the reduced matrix problem $\tilde{A} = U\Sigma\tilde{V}^T$
7: Compute the singular vectors $V = \tilde{Q} \times \tilde{V}$

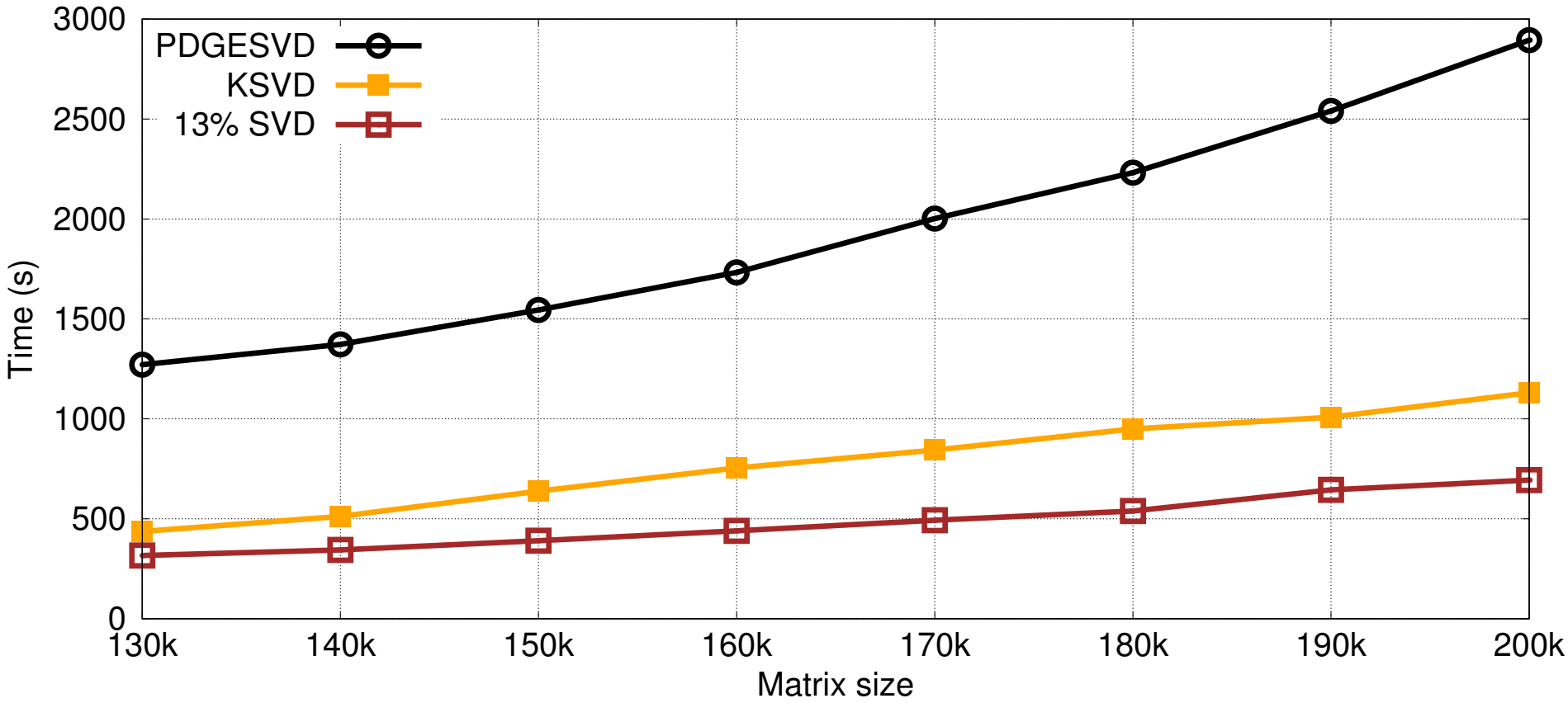
[5] David Keyes, Hatem Ltaief, Yuji Nakatsukasa, And Dalal Sukkari: Toward High Performance Partial Spectrum Computations for the Symmetric Eigensolver and the SVD using QDWH, SISC, (to be submitted)

Algorithmic Complexity

	Standard SVD	QDWH-based **Full** SVD	QDWH-based **Partial** SVD
Algorithmic complexity	$17n^3$	$52n^3$	QDWH: (4+1/3)n^3 x $\#it_{Chol}$ QR and GEMM: $4/3n^3$ + $2sn^2$ + $2ns^2$ SVD: $17s^3$

Where, n is the matrix size, and s is the number of the selected singular values/vectors ($s \ll n$).
Assuming $s \ll n$ and three iterations to get the polar factor from QDWH, the total number of operations is $14n3$.

Performance Comparison 128x288 (proc grid)



[4X, 2X] against [ScaLAPACK PDGESVD, KSVD], respectively.

Outline

1. Motivation
2. State-of-the-Art Approach
3. QDWH-based Polar Decomposition
4. Algorithmic Designs of QDWH
5. Application to Full/Partial SVD
6. **Application to Partial EIG**
7. Application to Extreme Adaptive Optics
8. Conclusion and Future Work

Extension to Partial EIG

$$A = V_{1:s} \Lambda_{1:s} V_{1:s}^{\top}$$

$$A \in \mathbb{R}^{n \times n}$$

where:
s is the number of the eigenvalues values, Λ is the matrix of the eigenvalues, and
V is the matrix of the corresponding eigenvectors.

QDWHpartial-EIG

Algorithm 4 QDWHpartial-EIG.

1: Compute a, b using Lanczos iterations
2: Shift the matrix $AA = (A - (a+b)/2 * Id)/(b-a) * 2$
3: Compute the polar decomposition $AA = U_p H$ using QDWH
4: Calculate $[Q\ R] = QR(U_p + Id)$
5: Find the index $ind = min(find(abs(diag(R)) < threshold))$
6: Extract $\tilde{Q} = Q(:, ind : end)$
7: Reduce the original matrix problem $\tilde{A} = \tilde{Q}' \times A \times \tilde{Q}$
8: Compute the EIG of the reduced matrix problem $\tilde{A} = V \Lambda \tilde{V}^T$
9: Compute the eigenvectors $V = \tilde{Q}^T \times \tilde{V}$

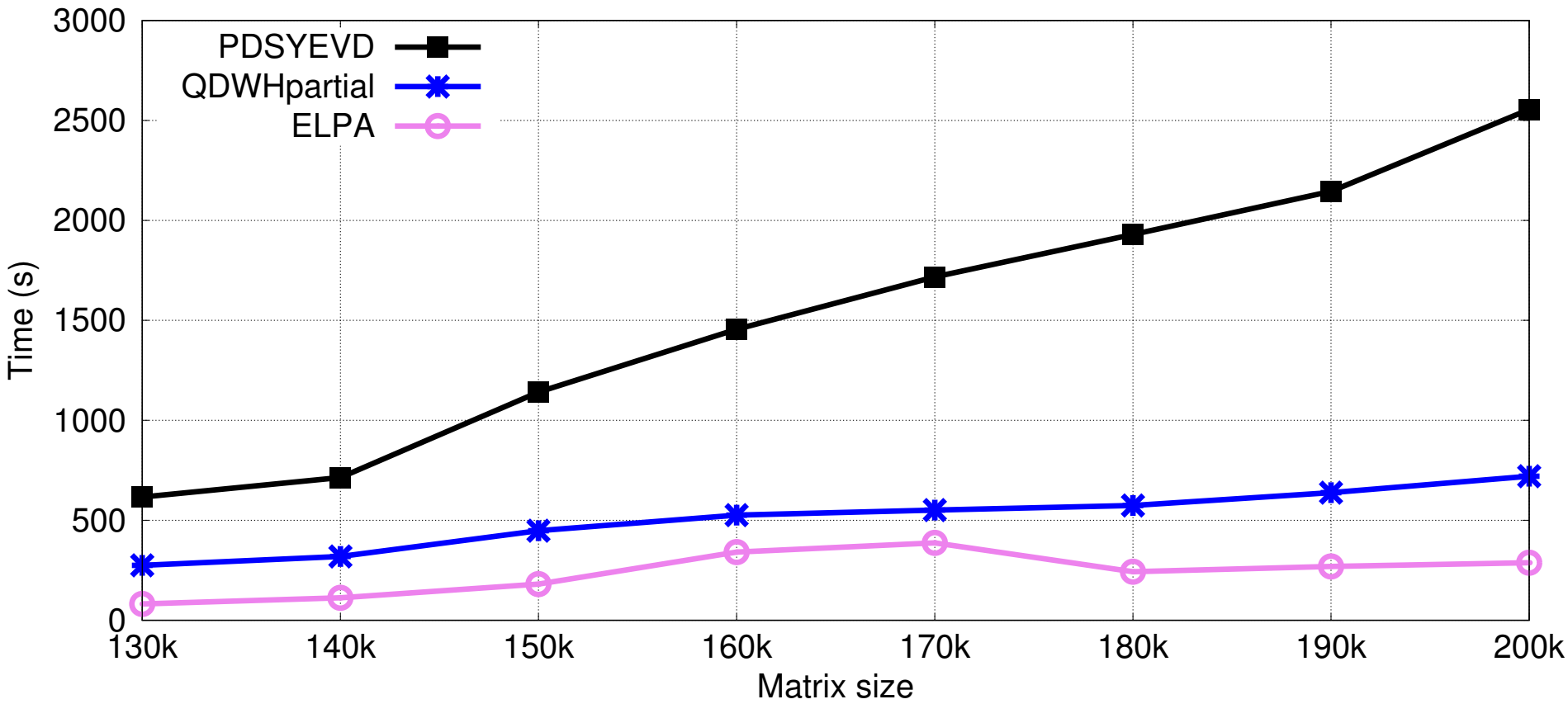
[5] David Keyes, Hatem Ltaief, Yuji Nakatsukasa, And Dalal Sukkari: Toward High Performance Partial Spectrum Computations for the Symmetric Eigensolver and the SVD using QDWH, SISC, (to be submitted)

Algorithmic Complexity

	ELPA EIG	QDWH-based **Full** EIG	QDWH-based **Partial** EIG
Algorithmic complexity	$(6 + 2/3)n^3$	$47n^3$	QDWH: (4+1/3)n^3 x $\#it_{Chol}$ QR and GEMM: $4/3n^3$ + $2sn^2$ + $2ns^2$ EIG: $6s^3$

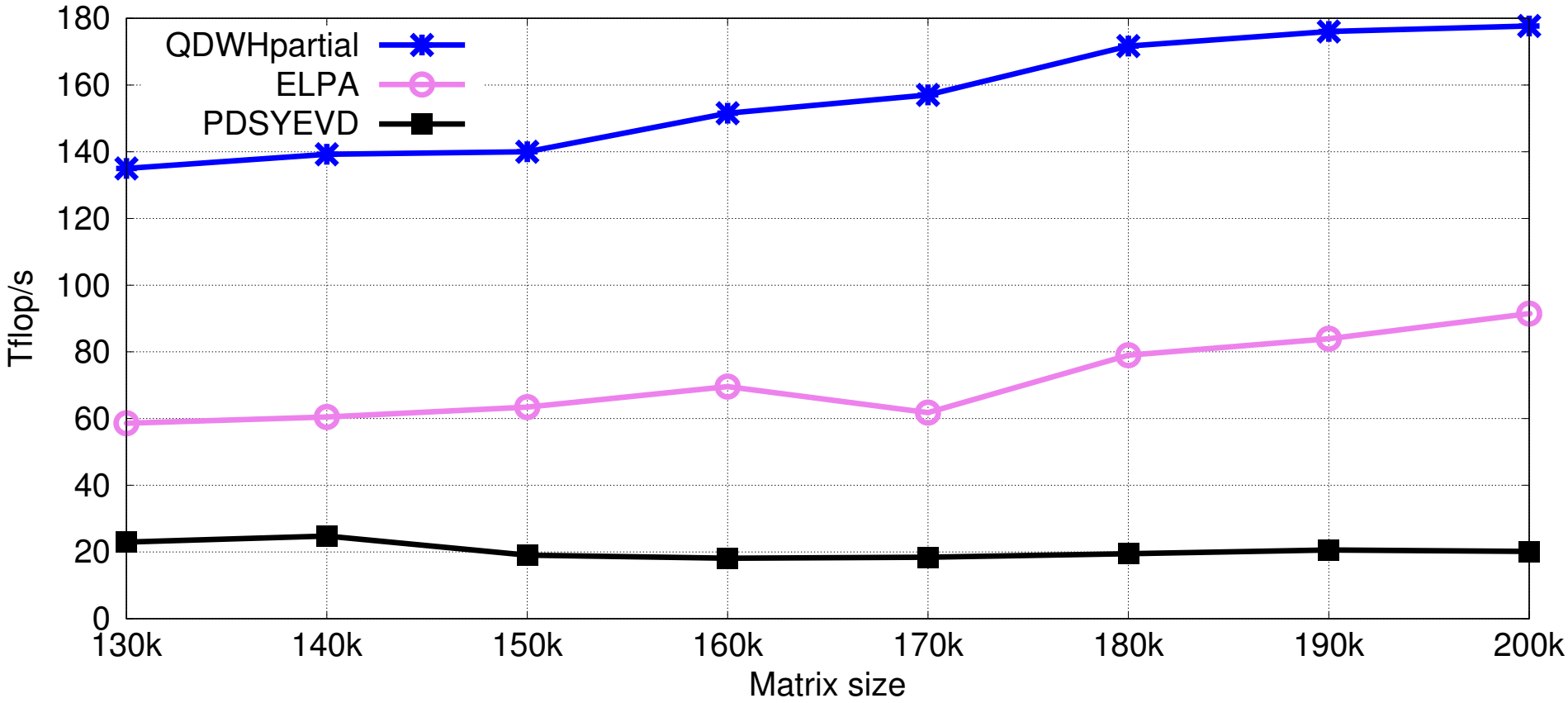
Where, n is the matrix size, and s is the number of the selected eigenvalues/vectors ($s \ll n$). Assuming $s \ll n$ and three iterations to get the polar factor from QDWH, the total number of operations is $14n3$.

Performance Comparison 128x288 (proc grid)



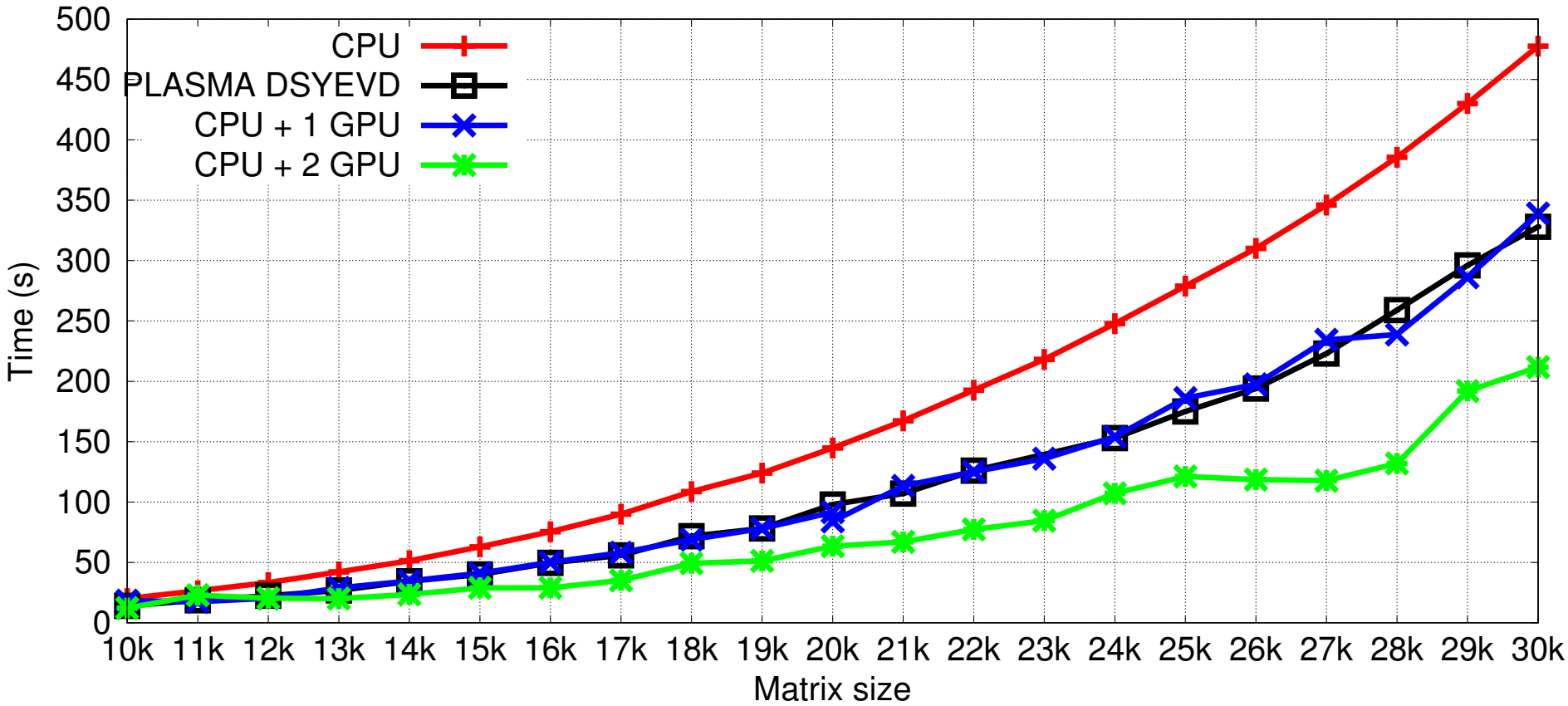
Elpa is [6.0X, 2.6X] against [ScaLAPACK PDSYEVD, QDWHpartial-EIG], respectively. QDWHpartial-EIG 3.9X against ScaLAPACK PDSYEVD.

Performance Comparison 128x288 (proc grid)



[8.9X, 2.0X] against [ScaLAPACK PDSYEVD, ELPA], respectively.

Let's Use Task-Based Asynchronous Execution (in Time)



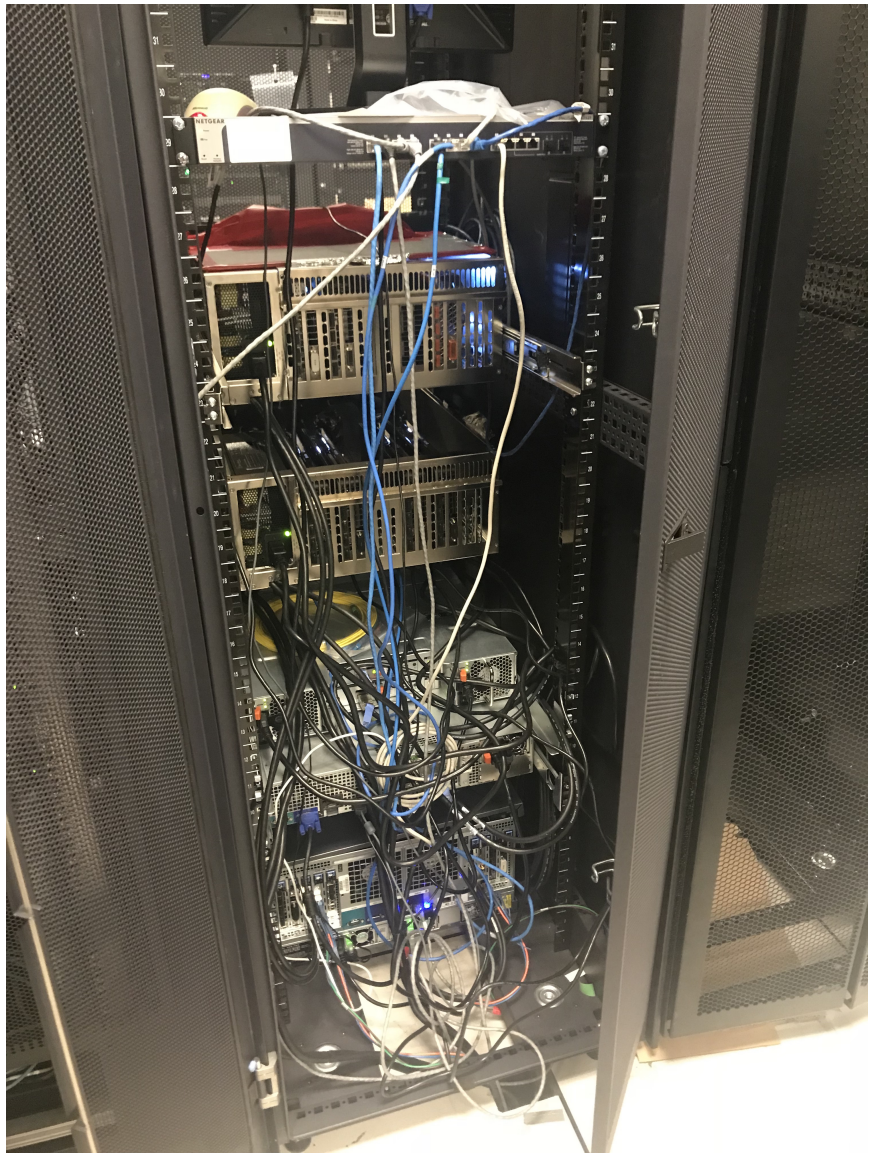
Outline

1. Motivation
2. State-of-the-Art Approach
3. QDWH-based Polar Decomposition
4. Algorithmic Designs of QDWH
5. Application to Full/Partial SVD
6. Application to Partial EIG
7. **Application to Extreme Adaptive Optics**
8. Conclusion and Future Work

The Subaru Telescope

- The pseudo-inverse using QDWHpartial-SVD code is deployed on-sky for the Subaru telescope during observation nights
- Represents a flagship telescope of the National Astronomical Observatory of Japan
- Carries a 8.2-meter (320in) diameter telescope
- Contains a high-contrast imaging system for directly imaging exoplanets
- Operates perhaps the most advanced HPC facility in computational astronomy
- Lives at the Mauna Kea Observatory in Hawaii

(Perhaps) The Highest in Altitude GPU System Recorded at 14,000 feet!



Environment Settings

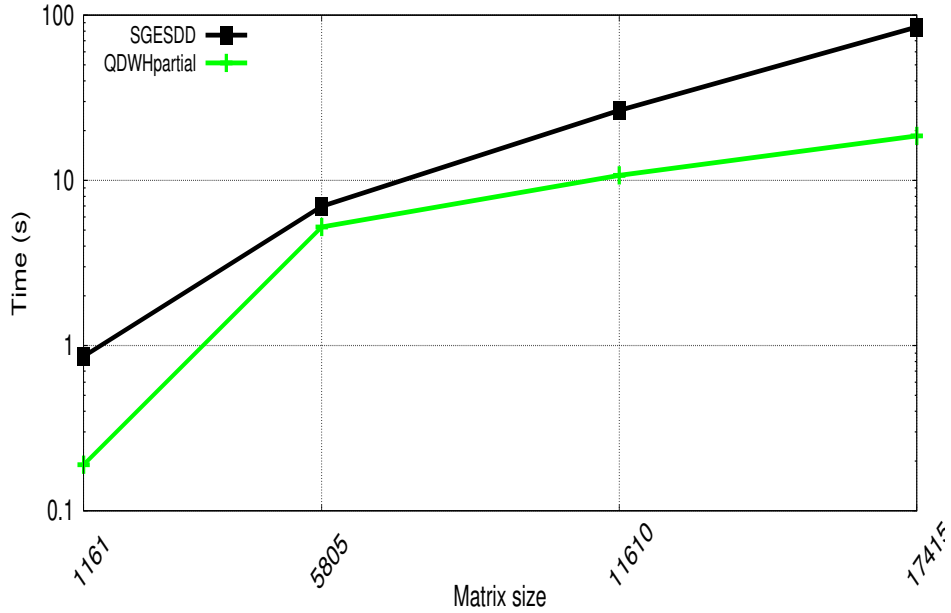
Software:

- GCC compilers
- `MAGMA` v2.3 and `CUDA` v9.0 (including `cuBLAS`)
- Single precision (SP) arithmetic is used
- Ill-conditioned matrices generated using *SLATMS* MAGMA routine

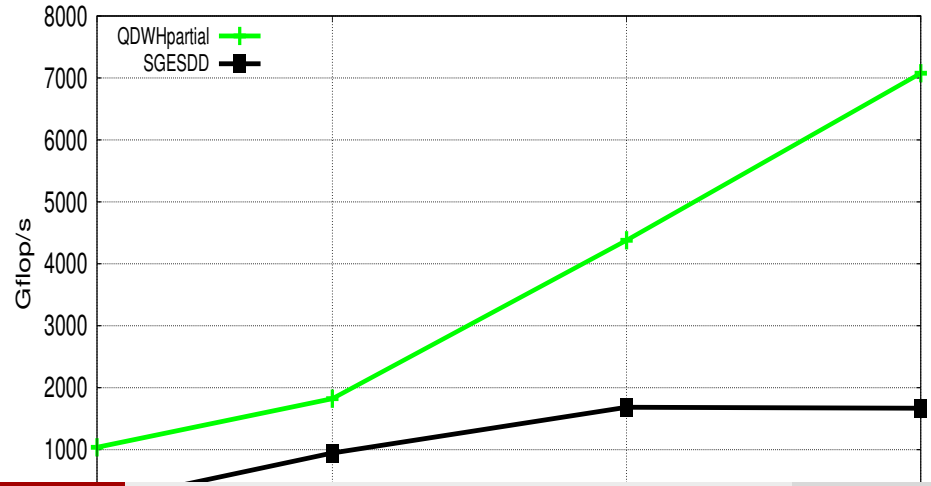
Hardware:

- The `K80` GPU:
  - 12GB of memory,
  - Two-socket 14-core system
  - Intel Broadwell system with 128GB of main memory
- The `P100` and `V100` GPUs:
  - 16GB of memory
  - Two-socket 16-core system
  - Intel Haswell systems with 128GB of main memory

Real Observational Datasets Using V100 GPU



(a) In Time.



Outline

1. Motivation
2. State-of-the-Art Approach
3. QDWH-based Polar Decomposition
4. Algorithmic Designs of QDWH
5. Application to Full/Partial SVD
6. Application to Partial EIG
7. Application to Extreme Adaptive Optics
8. **Conclusion and Future Work**

Conclusion

- Various algorithmic strategies have been employed to make QDWH-based algorithms attractive
- The QDWH-SVD achieves gains up to **3.6X** compared to the state-of-the-art solver
- The QDWHpartial-SVD achieves **6X** and **2X** speedup against ScaLAPACK-PDGESVD and QDWH-SVD, respectively
- QDWHpartial-SVD in the computational astronomy application provides up to **4X** speedup
- Preliminary implementation of QDWHpartial-EIG achieves up to **2X** speedup against PLASMA eigensolver

Future Research Directions

- Use the fine-grained task model to build QDWH-EIG and QDWHpartial-EIG on distributed-memory systems equipped with GPUs
- Use the QDWHpartial-EIG in the image processing and material science applications

Thank you ☺