CRAY

Performance Impact of Rank-Reordering on Advanced Polar Decomposition Algorithms

CUG 2018

Aniello Esposito, Cray Inc.
David Keyes, Hatem Ltaief, Dalal Sukkari, KAUST

Abstract

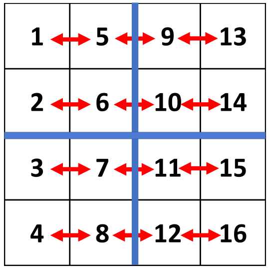
- **Investigated impact of MPI rank-reordering and BLACS grid topology on advanced polar decompositions of dense matrices.**
  - Naturally assumed to be compute bound, but in the strong scaling limit the algorithms suffer from communication and load balancing.
    - Modifying grid topology and rank-reordering improves both.
  - QDWH and ZOLOPD algorithms based on ScaLAPACK.
- **Used profiling to identify affected code regions.**
  - Focused on QDHW which benefits more from these modifications.

Context of this Work

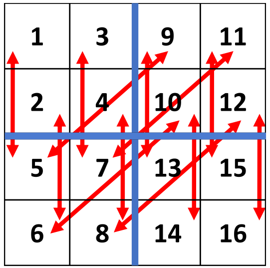
- **Cray Center of Excellence at King Abdullah University of Science and Technology (KAUST).**
  - High-level collaborations with principal investigators in areas ranging from traditional HPC (Combustion, Linear Algebra, …) to Analytics and Machine Learning.
  - Maintained by the Cray EMEA Research LAB (CERL)
- **Hierarchical Computations on Manycore Architectures**
  - Is the project from which the present work arose.
  - Extreme Computing Research Lab at KAUST.

Rank-Reordering Feature of `cray-mpich`

- (a) column-major global rank ordering used internally
- (b) SMP-style rank ordering used by default on the Cray XC system. Red arrows correspond to the ones in (a).



(a)



(b)

```
MPICH_RANK_ORDER
1,  5,  2,  6
3,  7,  4,  8
9,  13, 10, 14
11, 15, 12, 16
```
(c)

- (c) Rank reo-rdering file yielding the placement shown in (a).
  - The `cray-mpich` library allows to override the default MPI rank placement scheme by means of the `MPICH_RANK_REORDER_METHOD` environment variable.

ScaLAPACK Grid Topology

- Basic Linear Algebra Communication Subroutines
  - `Cblacs_gridinit( &ictxt, "R", P, Q );`

(P,Q) = (2,8)

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15

(P,Q) = (4,4)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Polar Decomposition

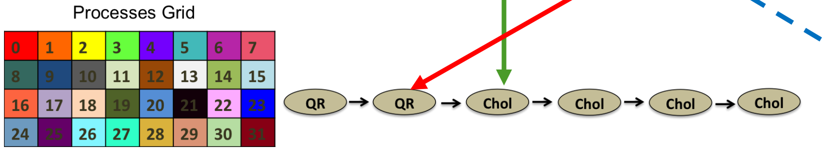
- **Of the form *A=UP*, where *U* is a unitary matrix and *P* is a positive-semidefinite Hermitian matrix**
  - Can be computed with Heron's method
  - More advanced (inverse-free) algorithms are
    - QR-based Dynamically Weighted Halley method (QDWH)
    - Zolotarev rational functions (ZOLOPD).
- **PD is the first computational step toward solving symmetric eigenvalue problems and the singular value de- composition.**

Polar Decomposition

QDWH

$$X_0 = A/\alpha,$$
$$\begin{bmatrix} \sqrt{c_k} X_k \\ I \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R,$$
$$X_{k+1} = \frac{b_k}{c_k} X_k + \frac{1}{\sqrt{c_k}} \left(a_k - \frac{b_k}{c_k} \right) Q_1 Q_2^\top, \quad k \geq 0.$$

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

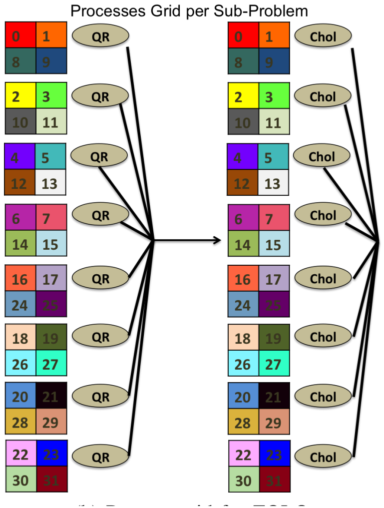


Maximum of 6 iterations

ZOLOPD

$$\begin{bmatrix} X \\ \sqrt{c_{2j-1}} I \end{bmatrix} = \begin{bmatrix} Q_{j1} \\ Q_{j2} \end{bmatrix} R_j,$$
$$Z_{2r+1}(X;\ell) = X + \Sigma_{j=1}^{r} \frac{a_j}{\sqrt{c_{2j-1}}} Q_{j1} Q_{j2}^*.$$

Sub-problems joined in a large matrix.



	QDWH	Successive ZOLO-PD	Independent ZOLO-PD
# QR-based iterations	2	8	1
# Cholesky-based iterations	4	8	1
Algorithmic complexity	$33n^3$	$100n^3$	$15\ n^3$
Memory footprint	$6n^2$	$6n^2$	$48n^2$

Simulation Setup

- **Dedicated Cray XC system**
  - Featuring dual Intel Broadwell processor compute nodes with
  - 128GB DDR4 memory each
  - Running with Moab/Torque+ALPS.
    - Number of cores and base clock frequencies are not uniform across compute nodes.
    - Only 32 cores per node were used with a frequency capped to 2.1GHz for the experiments.
    - The codes were built with the Intel Compiler 17.0.1.132 and the corresponding MKL libary for the basic linear algebra computations
- **Matrices considered in QDWH and ZOLOPD**
  - Range from 71680 to 122880 in steps of 10240 and are factorized on 200, 400, and 800 compute nodes using one MPI rank per core, where the MPI ranks are arranged by ScaLAPACK in a row-major order on a P × Q grid.

Simulation Setup

- **Rank-Reordering**
  - Both row-major and column-major global rank re-orderings have been considered with two different on- node orderings.

- **Only a few grid topologies P×Q are possible with a compatible rank-reordering.**

Command	Label
`<no reordering>`	0
`grid_order -R -c 8,4 -g P,Q`	1
`grid_order -R -c 4,8 -g P,Q`	2
`grid_order -C -c 8,4 -g P,Q`	3
`grid_order -C -c 4,8 -g P,Q`	4

ZOLOPD

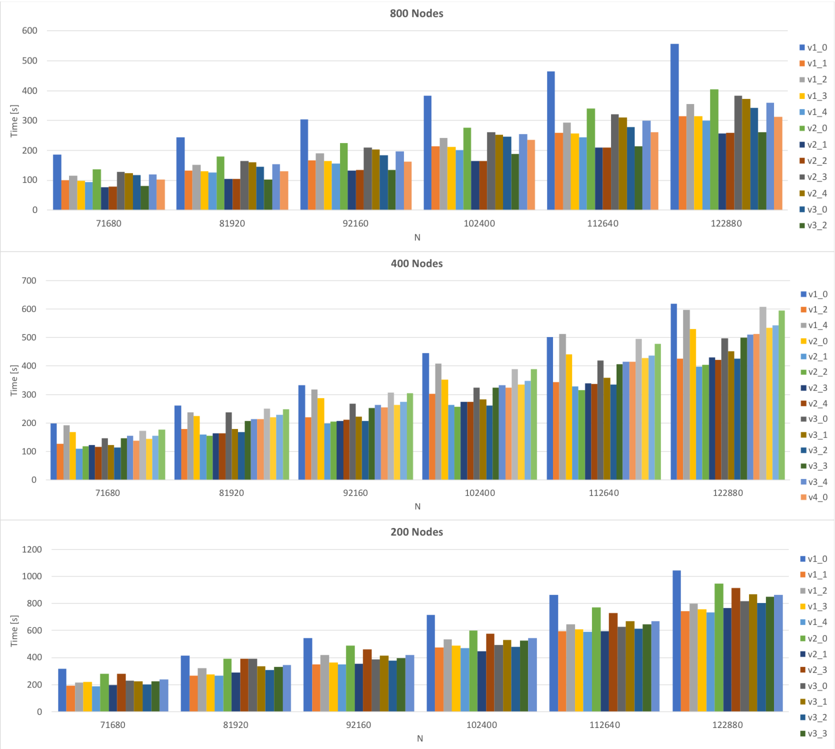
Nodes	Ranks	P	Q	p	q	$R = P/Q$	$r = p/q$	Reorder	Label
800	25600	80	320	40	80	0.25	0.5	1	v1_0
								1	v1_1
								2	v1_2
								3	v1_3
								4	v1_4
		80	320	20	160	0.25	0.125	0	v2_0
								2	v2_2
								4	v2_4
		160	160	40	80	1	0.5	0	v3_0
								1	v3_1
								2	v3_2
								3	v3_3
								4	v3_4
400	12800	80	160	40	40	0.5	1	0	v1_0
								1	v1_1
								2	v1_2
								3	v1_3
								4	v1_4
		64	200	32	50	0.32	0.64	0	v2_0
200	6400	40	160	20	40	0.25	0.5	0	v1_0
								2	v1_2
								4	v1_4
		40	160	10	80	0.25	0.125	0	v2_4
		80	80	20	40	1	0.5	0	v3_0
								2	v3_2
								4	v3_4

QDWH

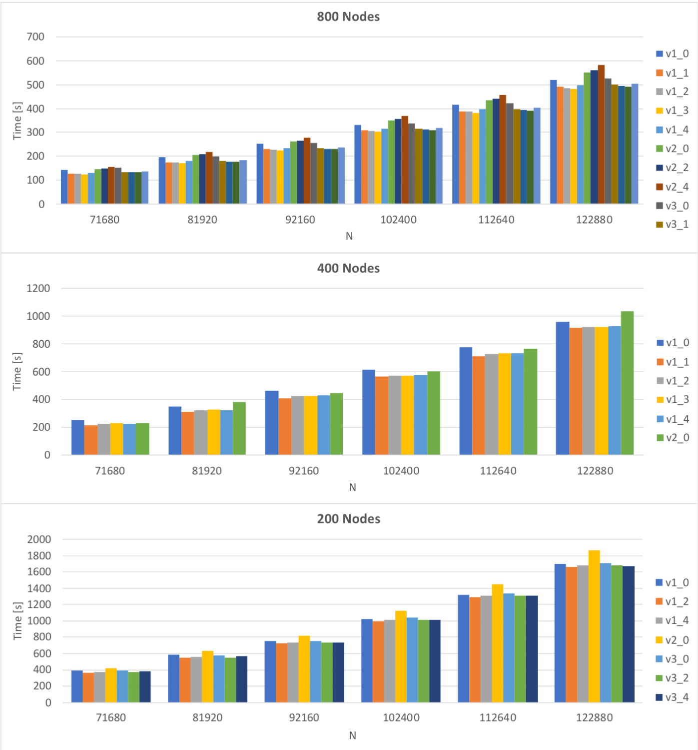
Nodes	Ranks	P	Q	$R = P/Q$	Reorder	Label
800	25600	160	160	1	0	v1_0
					1	v1_1
					2	v1_2
					3	v1_3
					4	v1_4
		128	200	0.64	0	v2_0
					1	v2_1
					2	v2_2
					3	v2_3
					4	v2_4
		100	256	0.39	0	v3_2
					2	v3_3
					4	v3_4
		50	512	0.098	0	v4_0
400	12800	100	128	0.78	0	v1_0
					2	v1_2
					4	v1_4
		80	160	0.5	0	v2_0
					1	v2_1
					2	v2_2
					3	v2_3
					4	v2_4
		64	200	0.3	0	v3_0
					1	v3_1
					2	v3_2
					3	v3_3
					4	v3_4
		32	400	0.08	0	v4_0
					1	v4_1
					2	v4_2
					3	v4_3
					4	v4_4
200	6400	80	80	1	0	v1_0
					1	v1_1
					2	v1_2
					3	v1_3
					4	v1_4
		64	100	0.64	0	v2_0
					1	v2_1
					3	v2_2
		32	200	0.16	0	v3_0
					1	v3_1
					2	v3_2
					3	v3_3
					4	v3_4

Complete Set of Simulation Results

QDWH

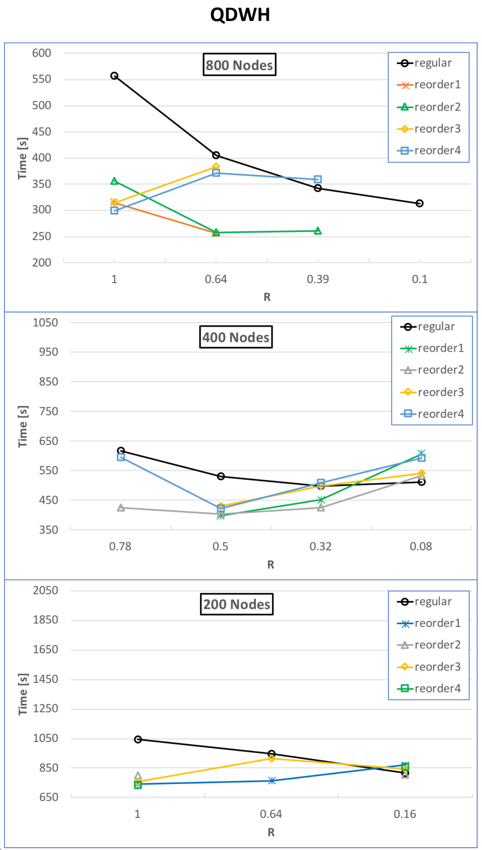


ZOLOPD



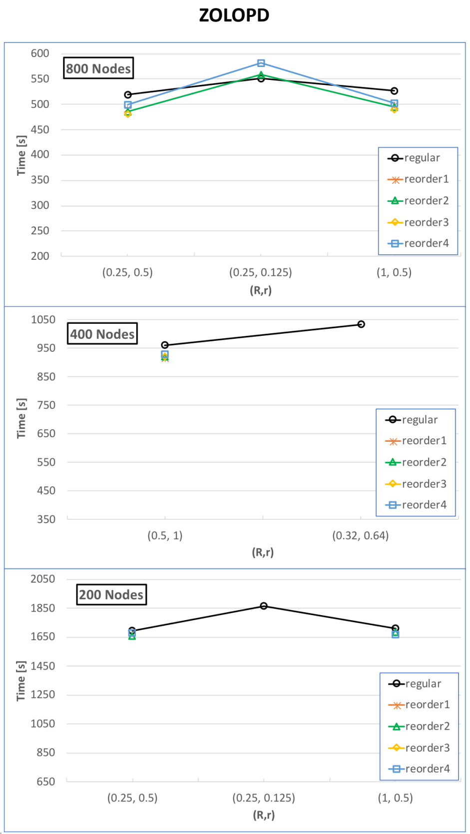
Impact of Topology and Rank-Reordering

- **QDWH (largest matrix)**
  - Improvement of the total execution time for the QDWH algorithm when lowering the ratio R without reordering.
  - Row-major in this case, is more beneficial than column-major especially on a large amount of nodes.
    - Consistent with the ScaLAPACK grid topology
  - On-node reordering -c 4,8 yields a better performance than -c 8,4 only for R = 1.



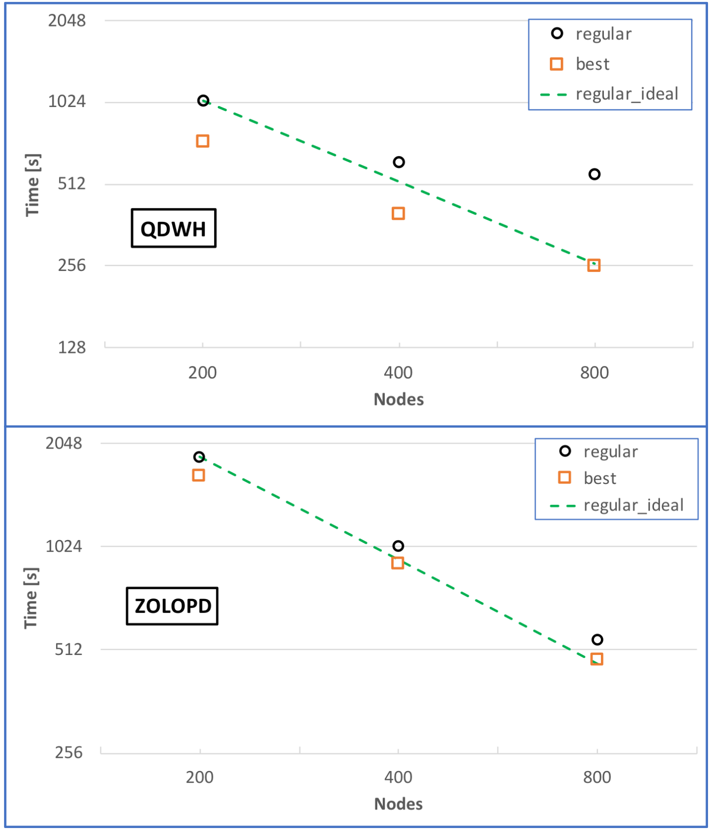
Impact of Topology and Rank-Reordering

- **ZOLOPD (largest matrix)**
  - Does not manifest a comparable improvement to QDWH when rank reordering or different grid topologies are used.
  - This is observed for every node count.
  - The reason is assumed to be in the sub-problem structure and has to be further investigated.



Impact of Topology and Rank-Reordering

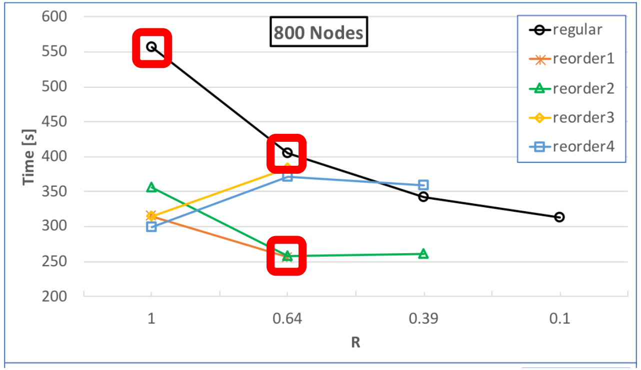
- **Strong Scaling**
  - Is improved in both cases when using the best combination instead of the least performant SMP-style ordering.
  - The best solver times approach the ideal scaling curve of the least performant SMP-style ordering in the strong scaling limit.



Profiling Analysis of QDWH

- **Focusing on three data points.**
  - Pure topology change notably improves the QR and Cholesky decompositions, while shifting the focus of the work towards QR.
  - Using the most performant combination of rank reordering and grid topology further improves the individual decompositions and levels out the relative amount of work between the two.

	R=1	R=0.64	reorder1, R=0.64
Total	100	100 (75.44)	100 (47.96)
Cholesky	49.29	34.39 (25.94)	41.84 (20.06)
QR	40.53	52.92 (39.92)	46.81 (22.45)
timeLi	6.97	8.99 (6.79)	5.44 (2.61)
timeFormH	2.61	2.88 (2.18)	4.19 (2.01)



Profiling Analysis of QDWH

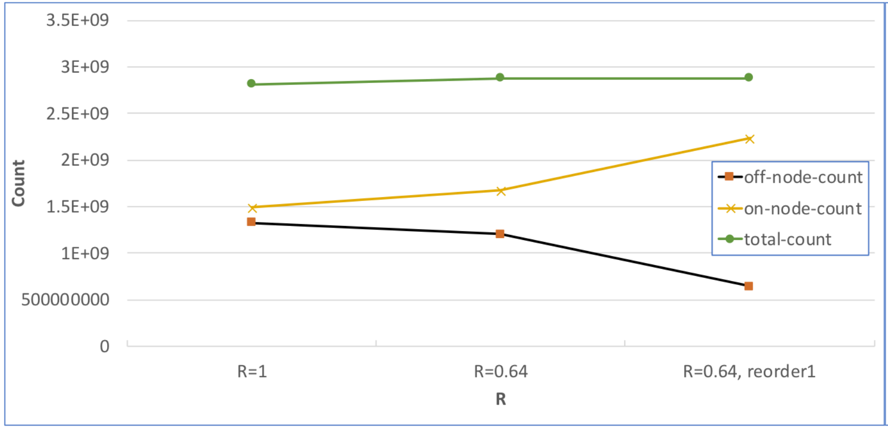
- **Focusing on three data points.**
  - Changing topology and reordering successively reduces the relative amount of communication time while keeping the dominant portion of communication time in `MPI_Recv` and synchronization in `MPI_Reduce`.
  - Gap between `MPI_Recv` and `MPI_Send` indicates that a large amount of time is spent in simply waiting in blocking receivers in point-to- point communication.

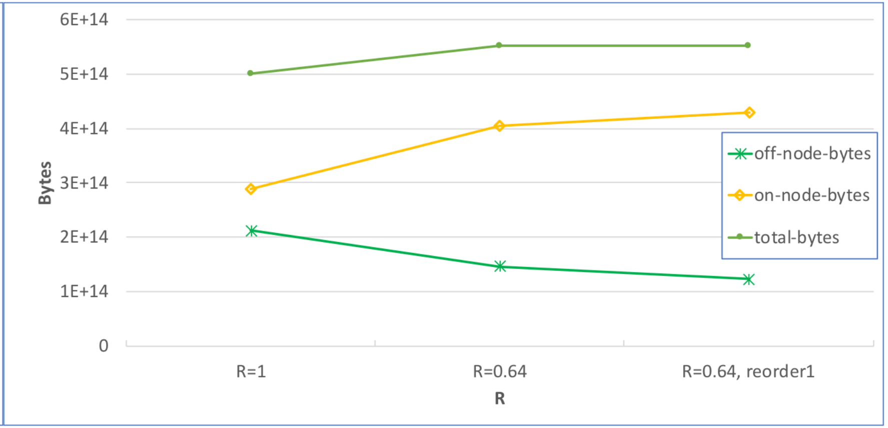
	R=1	R=0.64	reorder1, R=0.64
Total	100	100 (75.44)	100 (47.96)
Cholesky	49.29	34.39 (25.94)	41.84 (20.06)
QR	40.53	52.92 (39.92)	46.81 (22.45)
timeLi	6.97	8.99 (6.79)	5.44 (2.61)
timeFormH	2.61	2.88 (2.18)	4.19 (2.01)

	R=1	R=0.64	reorder1, R=0.64
Total	100	100 (75.44)	100 (47.96)
Total MPI	84.07	76.56 (57.75)	60.26 (28.9)
Recv	53.33	44.08 (33.25)	25.89 (12.42)
Bcast	3.44	3.34 (2.52)	3.32 (1.59)
Bcast(sync)	3.81	4.95 (3.73)	5.98 (2.87)
Reduce	9.55	8.18 (6.17)	3.08 (1.48)
Reduce(sync)	10.51	12.05 (9.09)	16.86 (8.09)
Send	3.01	3.65 (2.75)	4.87 (2.34)

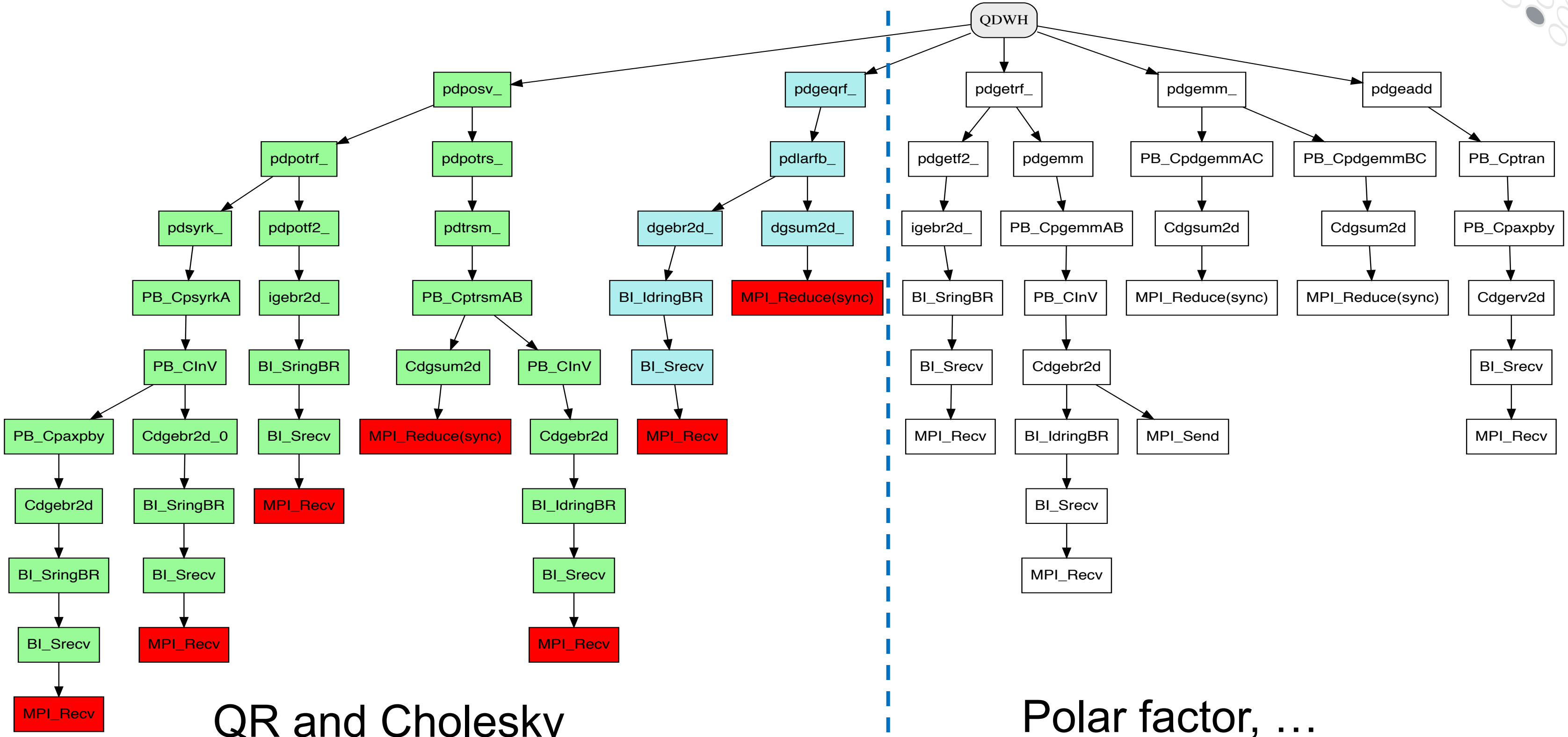
QDWH Message and Byte Counts

- **Changing the grid topology also implies an improvement of on/off-node message and traffic ratio in addition to algorithmic improvement.**
  - Rank-reordering provides a pure on/off-node ratio improvement.





QDWH Call-Tree



QR and Cholesky

Polar factor, …

Conclusions

- **A suitable combination of rank-reordering and grid topology can considerably improve the performance of dense linear algebra algorithms in the strong scaling limit.**
  - Here we considered advanced polar decompositions.
- **Profiling analysis reveals an improvement of point-to-point communication in particular in MPI_Revc.**
  - Analysis restricted to QDWH
  - ZOLO needs more investigation.

Legal Disclaimer

Information in this document is provided in connection with Cray Inc. products. No license, express or implied, to any intellectual property rights is granted by this document.

Cray Inc. may make changes to specifications and product descriptions at any time, without notice.

All products, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Cray hardware and software products may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Cray uses codenames internally to identify products that are in development and not yet publicly announced for release. Customers and other third parties are not authorized by Cray Inc. to use codenames in advertising, promotion or marketing and any use of Cray Inc. internal codenames is at the sole risk of the user.

Performance tests and ratings are measured using specific systems and/or components and reflect the approximate performance of Cray Inc. products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.

The following are trademarks of Cray Inc. and are registered in the United States and other countries: CRAY and design, SONEXION, URIKA and YARCDATA. The following are trademarks of Cray Inc.: CHAPEL, CLUSTER CONNECT, CLUSTERSTOR, CRAYDOC, CRAYPAT, CRAYPORT, DATAWARP, ECOPHLEX, LIBSCI, NODEKARE, REVEAL. The following system family marks, and associated model number marks, are trademarks of Cray Inc.: CS, CX, XC, XE, XK, XMT and XT. The registered trademark LINUX is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Other trademarks used on this website are the property of their respective owners.

Q&A

Aniello Esposito
esposito@cray.com