Communication avoiding algorithms: enlarged Krylov methods and preconditioners

Laura Grigori

Alpines

Inria Paris - LJLL, UPMC

Plan

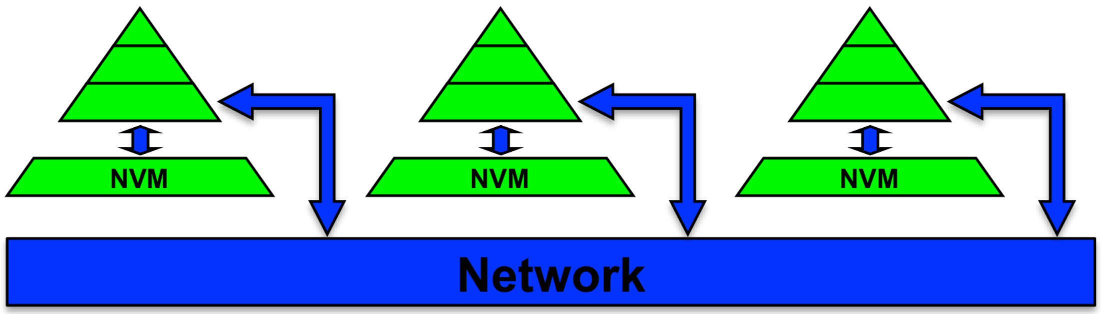
- Motivation
- Selected past work on reducing communication
- Brief overview of communication avoiding for dense linear algebra
  - LU, QR, Rank Revealing QR factorizations
  - Progressively implemented in ScaLAPACK, LAPACK
- Communication avoiding for sparse linear algebra
  - Krylov subspace methods
  - Preconditioners based on low rank corrections
- Conclusions

The communication wall

- Time to move data >> time per flop
  - Gap steadily and exponentially growing over time

- Annual improvements
  - Time / flop **59%**
  - Interprocessor bandwidth **26%**
  - Interprocessor latency **15%**
  - DRAM latency **5.5%**

- Performance of an application is less than 10% of the peak performance



Compelling numbers

DRAM bandwidth:

- Mid 90's ~ 0.2 bytes/flop – 1 byte/flop
- Past few years ~ 0.02 to 0.05 bytes/flop

DRAM latency:

- DDR2 (2007) ~ 120 ns — 1x
- DDR4 (2014) ~ 45 ns — 2.6x in 7 years
- Stacked memory ~ similar to DDR4

Time/flop

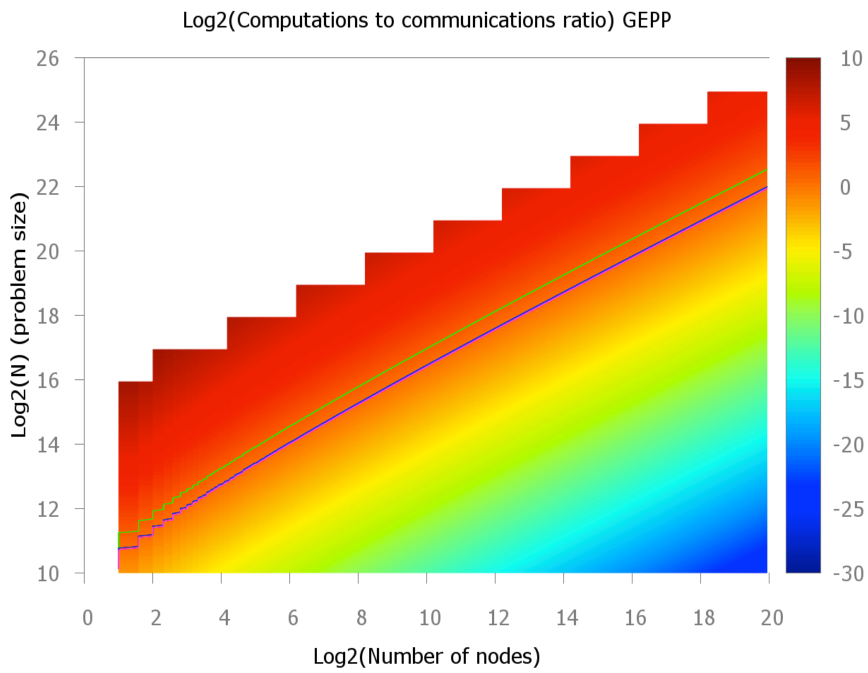
- 2006 Intel Yonah ~ 2GHz x 2 cores (32 GFlops/chip) — 1x
- 2015 Intel Haswell ~2.3GHz x 16 cores (588 GFlops/chip) — 18x in 9 years

Source: J. Shalf, LBNL

Approaches for reducing communication

- Tuning
  - Overlap communication and computation, at most a factor of 2 speedup

- Same numerical algorithm, different schedule of the computation
  - Block algorithms for NLA
    - Barron and Swinnerton-Dyer, 1960
    - ScaLAPACK, Blackford et al 97
  - Cache oblivious algorithms for NLA
    - Gustavson 97, Toledo 97, Frens and Wise 03, Ahmed and Pingali 00



- Same algebraic framework, different numerical algorithm
  - The approach used in CA algorithms
  - More opportunities for reducing communication, may affect stability

Communication Complexity of Dense Linear Algebra

- Matrix multiply, using $2n^3$ flops (sequential or parallel)
  - Hong-Kung (1981), Irony/Tishkin/Toledo (2004)
  - Lower bound on Bandwidth = Ω (#flops / $M^{1/2}$)
  - Lower bound on Latency = Ω (#flops / $M^{3/2}$)

- Same lower bounds apply to LU using reduction
  - Demmel, LG, Hoemmen, Langou 2008

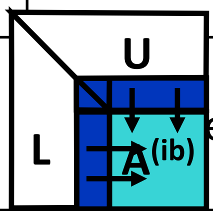
$$\begin{pmatrix} I & & -B \\ A & I & \\ & & I \end{pmatrix} = \begin{pmatrix} I & & \\ A & I & \\ & & I \end{pmatrix} \cdot \begin{pmatrix} I & & -B \\ & I & AB \\ & & I \end{pmatrix}$$

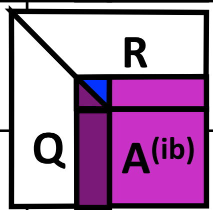
- And to almost all direct linear algebra [Ballard, Demmel, Holtz, Schwartz, 09]

2D Parallel algorithms and communication bounds

- If memory per processor = n^2 / P, the lower bounds on communication are #words_moved ≥ Ω (n^2 / $P^{1/2}$), #messages ≥ Ω ($P^{1/2}$)

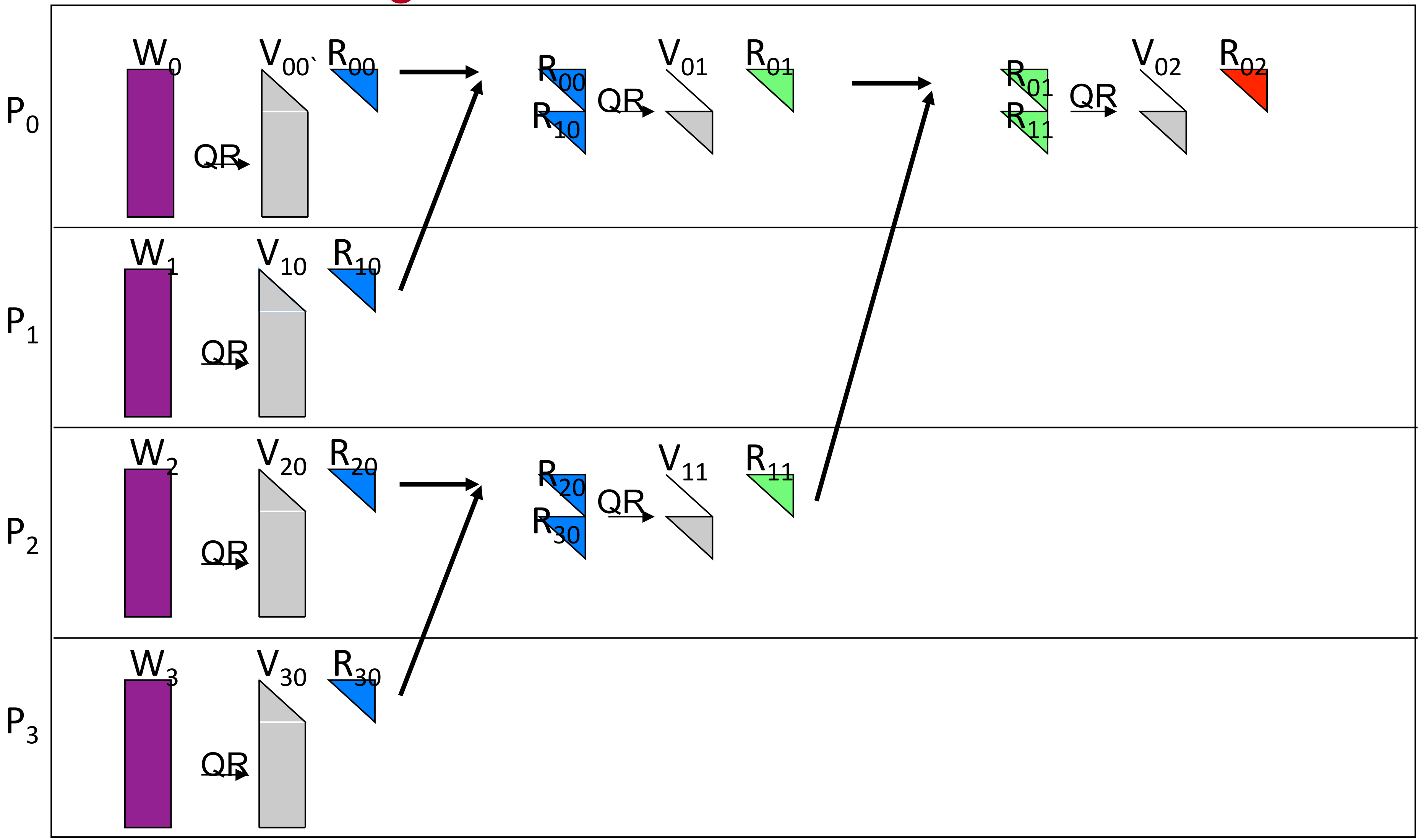
Algorithm	Minimizing #words (not #messages)	Minimizing #words and #messages
Cholesky	ScaLAPACK	ScaLAPACK
LU	ScaLAPACK es partial pivoting	[LG, Demmel, Xiang, 08] [Khabou, Demmel, LG, Gu, 12] uses tournament pivoting
QR	ScaLAPACK	[Demmel, LG, Hoemmen, Langou, 08] uses different representation of Q
RRQR	ScaLAPACK	[Demmel, LG, Gu, Xiang 13] uses tournament pivoting, 3x flops





- Only several references shown, block algorithms (ScaLAPACK) and communication avoiding algorithms
- CA algorithms exist also for SVD and eigenvalue computation

TSQR: QR factorization of a tall skinny matrix using Householder transformations



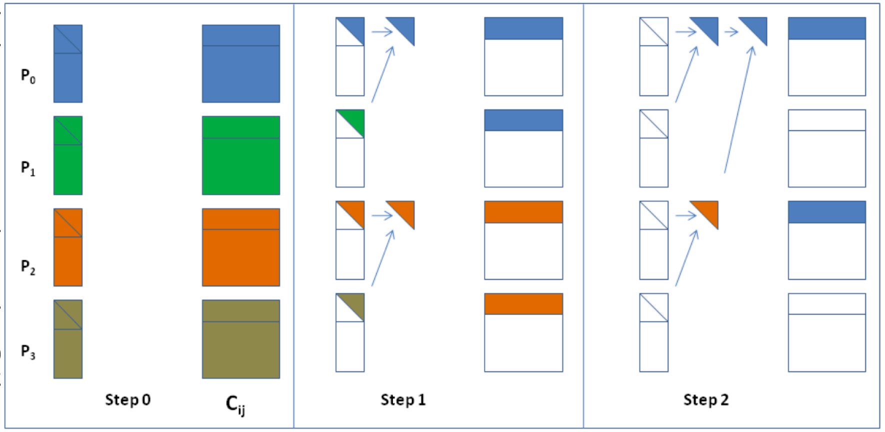
J. Demmel, LG, M. Hoemmen, J. Langou, 08

References: Golub, Plemmons, Sameh 88, Pothen, Raghavan, 89, Da Cunha, Becker, Patterson, 02

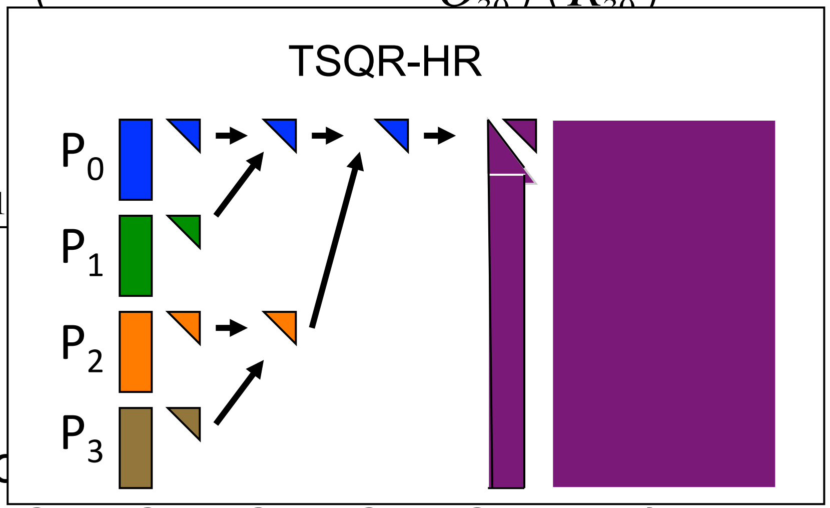
Algebra of TSQR

Parallel: $W = \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \end{bmatrix} \begin{matrix} \to \\ \to \\ \to \\ \to \end{matrix} \begin{matrix} R_{00} \\ R_{10} \\ R_{20} \\ R_{30} \end{matrix} \begin{matrix} \searrow R_{01} \searrow \\ \nearrow \\ \searrow R_{11} \nearrow \\ \nearrow \end{matrix} R_{02}$

$$W = \begin{pmatrix} W_0 \\ W_1 \\ W_2 \\ W_3 \end{pmatrix} = \begin{pmatrix} Q_{00}R_{00} \\ \hline Q_{10}R_{10} \\ \hline Q_{20}R_{20} \\ \hline Q_{30}R_{30} \end{pmatrix} = \begin{pmatrix} Q_{00} & & & \\ \hline & Q_{10} & & \\ \hline & & \ddots & \\ \hline & & & Q_{30} \end{pmatrix} \cdot \begin{pmatrix} \cdots \\ R_{30} \end{pmatrix}$$



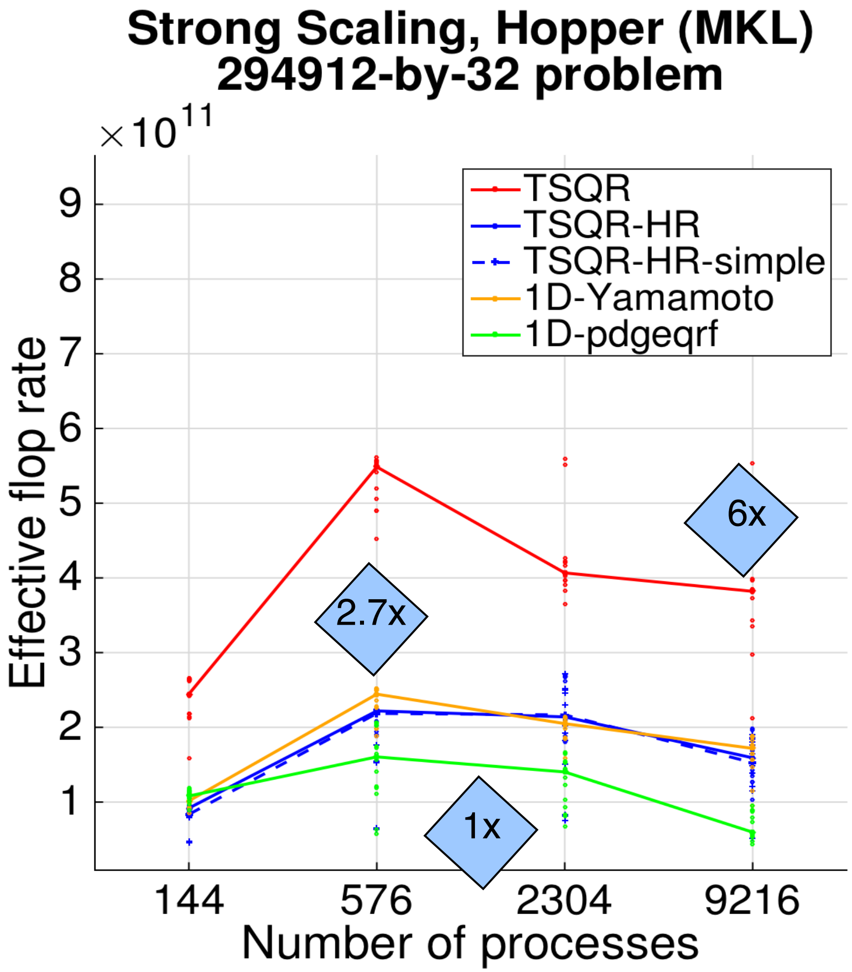
$$\begin{pmatrix} R_{00} \\ R_{10} \\ \hline R_{20} \\ R_{30} \end{pmatrix} = \begin{pmatrix} Q_{01}R_{01} \\ \hline Q_{11}R_{11} \end{pmatrix} = \begin{pmatrix} Q_{01} & \cdots \end{pmatrix}$$

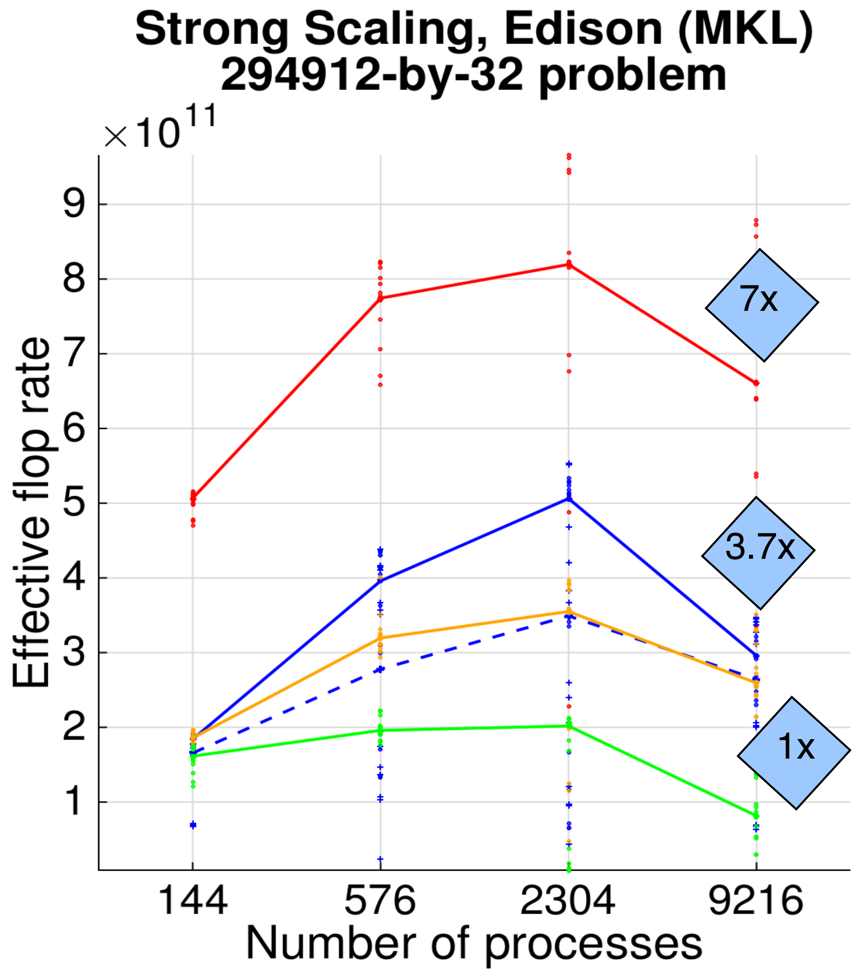


Q is represented implic

Output: $\{Q_{00}, Q_{10}, Q_{00}, Q_{20}, Q_{30}, Q_{01}, Q_{11}, Q_{02}, R_{02}\}$

Strong scaling





- Hopper: Cray XE6 (NERSC) – 2 x 12-core AMD Magny-Cours (2.1 GHz)
- Edison: Cray CX30 (NERSC) – 2 x 12-core Intel Ivy Bridge (2.4 GHz)
- Effective flop rate, computed by dividing $2mn^2 - 2n^3/3$ by measured runtime

Ballard, Demmel, LG, Jacquelin, Knight, Nguyen, and Solomonik, 2015.

Plan

- Motivation
- Selected past work on reducing communication
- Brief overview of communication avoiding for dense linear algebra
  - LU, QR, Rank Revealing QR factorizations
  - Progressively implemented in ScaLAPACK, LAPACK
- Communication avoiding for sparse linear algebra
  - Krylov subspace methods
  - Preconditioners based on low rank corrections
- Conclusions

Krylov subspace solvers

Solve $Ax = b$ by finding a sequence $x_1, x_2, \dots, x_k$ that minimizes some measure of error over the corresponding spaces

$$x_0 + \mathcal{K}_i(A, r_0), \quad i = 1, \dots, k$$

.

They are defined by two conditions:

1. Subspace condition: $x_k \in x_0 + \mathcal{K}_k(A, r_0)$
2. Petrov-Galerkin condition: $r_k \perp \mathscr{L}_k$

$$\iff (r_k)^t y = 0, \;\; \forall \;\; y \in \mathscr{L}_k$$

where

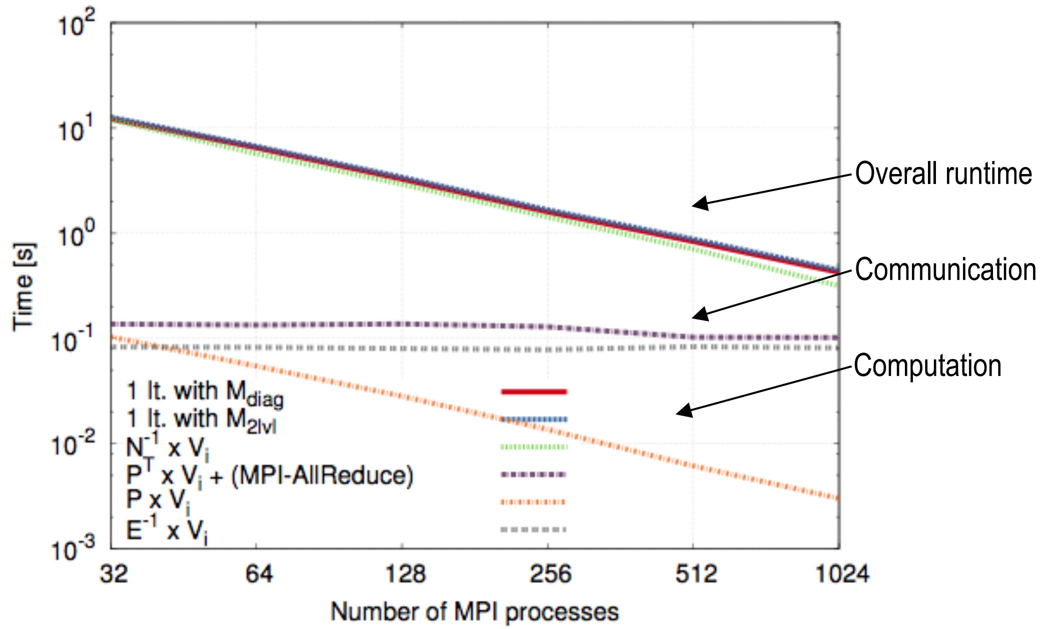
- x_0 is the initial iterate, r_0 is the initial residual,
- $\mathcal{K}_k(A, r_0) = span\{r_0, Ar_0, A^2 r_0, \dots, A^{k-1} r_0\}$ is the Krylov subspace of dimension k,
- $\mathscr{L}_k$ is a well-defined subspace of dimension k.

Challenge in getting efficient and scalable solvers

- A Krylov solver finds a solution x_k from $x_0 + K_k(A, r_0)$, where
$$K_k(A, r_0) = span\{r_0, A r_0, \ldots, A^{k-1} r_0\}$$
such that the Petrov-Galerkin condition $b - Ax_k \perp L_k$ is satisfied.
- Does a sequence of k SpMVs to get vectors $[x_1, \ldots, x_{k-1}]$
- Finds best solution x_k as linear combination of $[x_1, \ldots, x_{k-1}]$

- Each iteration requires
Sparse matrix vector product
-> point to point communication

Dot products for the
orthogonalization process
-> global synchronization



Map making, with R. Stompor, M. Szydlarski
Results obtained on Hopper, Cray XE6, NERSC

Ways to improve performance

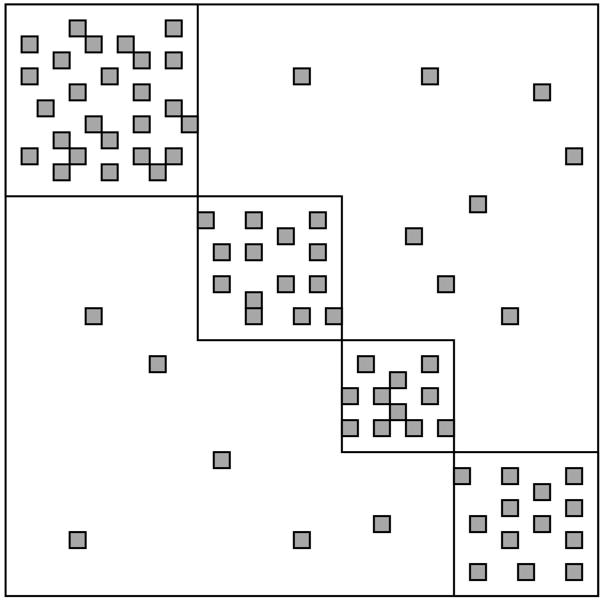
- Improve the performance of sparse matrix-vector product
- Improve the performance of collective communication

- Use preconditioners to decrease the number of iterations till convergence.

- Change numerics – enlarged Krylov methods
  - Decrease the number of iterations to decrease the number of global communications
  - Increase arithmetic intensity – compute sparse matrix-set of vectors product.

Enlarged Krylov subspaces

- Partition the matrix into t domains
- Split the initial residual into t vectors corresponding to the t domains



$$r_0 \rightarrow T(r_0) = \begin{bmatrix} * & 0 & & 0 \\ \vdots & \vdots & & \vdots \\ * & 0 & & 0 \\ 0 & * & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & * & & 0 \\ & & \ddots & \\ 0 & 0 & & * \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & * \end{bmatrix}$$

- Generate t new basis vectors, obtain an enlarged Krylov subspace

$$K_{t,k}(A, r_0) = span\left\{T(r_0), AT(r_0), \ldots, A^{k-1}T(r_0)\right\}$$

- Search for the solution of the system $Ax = b$ in $K_{t,k}(A, r_0)$

Properties of enlarged Krylov subspaces

- The Krylov subspace $K_k(A,r_0)$ is a subset of the enlarged one

$$K_k(A,r_0) \subset K_{t,k}(A,r_0)$$

- For all $k<k_{max}$, the dimensions of $K_{t,k}(A,r_0)$ and of $K_{t,k+1}(A,r_0)$ are strictly increasing by some number i_k and i_{k+1} respectively, where

$$t \geq i_k \geq i_{k+1} \geq 1$$

- The enlarged subspaces are increasing subspaces, yet bounded.

$$K_{t,1}(A,r_0) \subset \ldots \subset K_{t,k_{\max}-1}(A,r_0) \subset K_{t,k_{\max}}(A,r_0) = K_{t,k_{\max}+q}(A,r_0), \forall q > 0$$

- The solution of the system Ax=b belongs to the subspace $x_0+K_{t,k_{max}}$

LG, Moufawad, Nataf, 2014.

Enlarged Krylov subspace methods based on CG

Defined by the subspace $K_{t,k}$ and the following two conditions

$$1.\ \text{Subspace condition:} \qquad x_k \in x_0 + K_{t,k}$$

$$2.\ \text{Orthogonality condition:}\ r_k \perp K_{t,k}$$

- At each iteration the new approximate solution x_k is found by minimizing $\phi(x) = \frac{1}{2} x^T A x - b^T x$ over $x_0 + K_{t,k}$

$$\phi(x_k) = \min\left\{\phi(x), \forall x \in x_0 + K_{t,k}(A, r_0)\right\}$$

Convergence analysis

Given

- A is an SPD matrix, x^* is the solution of $Ax = b$
- $||\overline{e}_k||_A = ||x^* - \overline{x}_k||_A$ is the k^{th} error of CG
- $||e_k||_A = ||x^* - x_k||_A$ is the k^{th} error of enlarged methods
- CG converges in $\overline{K}$ iterations

Result

Enlarged Krylov methods converge in K iterations, where $K \leq \overline{K} \leq n$.

$$||e_k||_A = ||x^* - x_k||_A \leq ||\overline{e}_k||_A$$

Enlarged CG

Algorithm 1 Classic CG

$r_0 = b - Ax_0$
$p_1 = \frac{r_0}{\sqrt{r_0^t A r_0}}$
while $||r_{k-1}||_2 > \varepsilon ||b||_2$ **do**
  $\alpha_k = p_k^t r_{k-1}$
  $x_k = x_{k-1} + p_k \alpha_k$
  $r_k = r_{k-1} - A p_k \alpha_k$
  $p_{k+1} = r_k - p_k (p_k^t A r_k)$
  $p_{k+1} = \frac{p_{k+1}}{\sqrt{p_{k+1}^t A p_{k+1}}}$
end while

Algorithm 2 EK-CG

$R_0 = T(b - Ax_0)$
$P_1 =$ A-orthonormalize(R_0)
while $|| \sum_{i=1}^{t} R_k^{(i)} ||_2 < \varepsilon ||b||_2$ **do**
  $\alpha_k = P_k^t R_{k-1}$ ▷ $t \times t$
  $X_k = X_{k-1} + P_k \alpha_k$ ▷ $n \times t$
  $R_k = R_{k-1} - A P_k \alpha_k$ ▷ $n \times t$
  $P_{k+1} = A P_k - P_k (P_k^t A A P_k) - P_{k-1} (P_{k-1}^t A A P_k)$ ▷ $n \times t$
  $P_{k+1} =$ A-orthonormalize(P_{k+1})
end while
$x = \sum_{i=1}^{t} X_k^{(i)}$ ▷ $n \times 1$

#messages per iteration
- O(1) from SpMV+
- O(log P) from dot products

#messages per iteration
- O(1) from SpMV+
- O(log P) from block CGS + A-ortho

Reduction of number of search directions

- In CG we have the following relation

$$\alpha_k = P_k^T R_{k-1}$$

- To select only adding-value search directions we use the truncated SVD:

$$\alpha_k \approx U_k^+ \Sigma_k^+ W_k^+$$

- The new search directions are given by the relation:

$$\begin{aligned} X_k &= X_{k-1} + P_k \alpha_k & P_k^1 &\in \Re^{n \text{ x } rank(\alpha_k)} & &\leftarrow \text{size reduced} \\ &= X_{k-1} + \left(P_k U_k^+\right)\left(\Sigma_k^+ V_k^{+T}\right) & \alpha_k^1 &\in \Re^{rank(\alpha_k) \text{ x } t} & &\leftarrow \text{size reduced} \\ &= X_{k-1} + P_k^1 \alpha_k^1 & X_k, R_k &\in \Re^{n \text{ x } t} & &\leftarrow \text{size unchanged} \end{aligned}$$

- Idea adapted from Robbé and Sadkane (2006)

Test cases: boundary value problem

- Skyscrapper problem – SKY2D

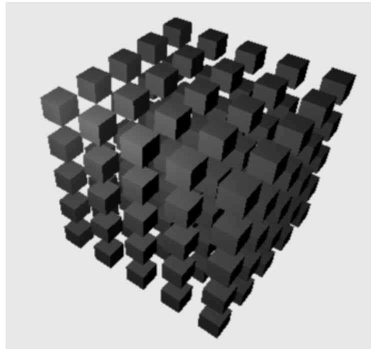
$$-div(\kappa(x)\nabla u) = f \quad in\,\Omega$$

$$u = 0 \quad on\,\partial\Omega_D$$

$$\frac{\partial u}{\partial n} = 0 \quad on\,\partial\Omega_N$$

$$\Omega = [0,1]^3, \partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$$

κ jumps from 1 to 10^3



discretized on a 2D or 3D grid

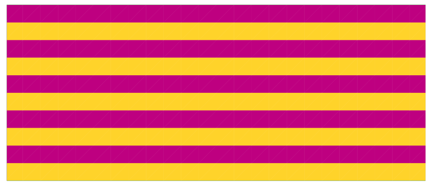
Test cases: linear elasticity

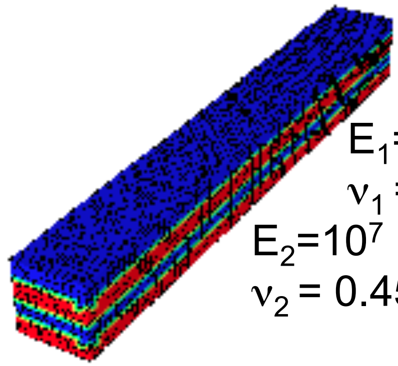
- Linear elasticity problems in 2D and 3D

$$\mathrm{div}\big(\sigma(u)\big) + f = 0 \qquad \text{on } \Omega$$

$$u = u_D \qquad \text{on } \partial\Omega_D$$

$$\sigma(u)\cdot n = g \qquad \text{on } \partial\Omega_N$$





where

$u \in R^d$ is the unknown displacement field

f is some body force

The Cauchy stress tensor σ(u) is given by Hooke's law,

$$\sigma(u) = 2\mu\varepsilon(u) + \lambda\cdot \mathrm{Tr}(\varepsilon(u))I$$

Material properties: Lame parameters λ and μ or alternatively Young's modulus E and Poisson's ratio v as

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{\mathrm{E}}{2(1+\nu)}$$

Numerical results

- Block Jacobi preconditioner (1024 blocks)
- Stopping criterion 10^{-6}
- Initial block size 32

		PCG		EK-CG		
	red. size	iter	error	iter	error	$\dim(\mathcal{K}_k^{\triangle})$
SKY2D	×	655	9.0e-08	57	7.3e-12	1824
	✓	655	9.0e-08	59	7.8e-12	1546
Ela3D100	×	870	3.5e-09	109	3.2e-11	3488
	✓	870	3.5e-09	116	5.3e-11	2384
Ela2D200	×	4551	1.2e-09	253	1.8e-10	8096
	✓	4551	1.2e-09	266	1.8e-10	6553

LG, Tissot, 2016.

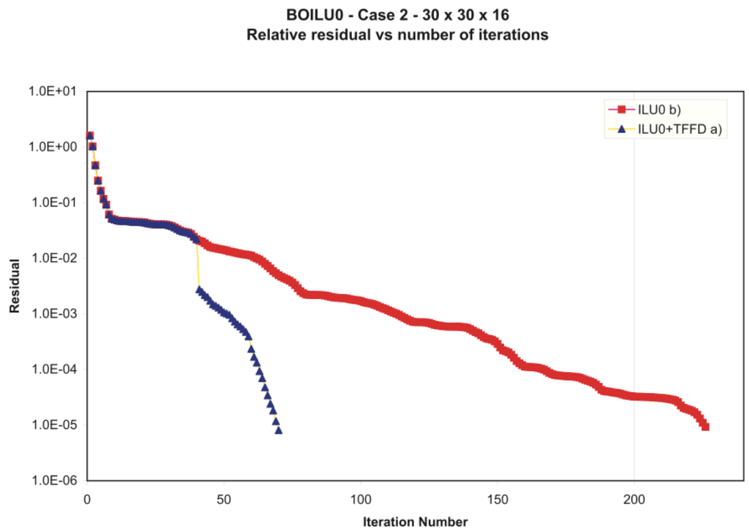
Challenge in getting scalable preconditioners

- Solve linear systems arising from large discretized systems of PDEs with strongly heterogeneous coefficients (high contrast, multiscale)

Darcy $\quad a(u,v) = \int_{\Omega} \kappa \, \nabla u \cdot \nabla v \; dx$

Elasticity $\quad a(u,v) = \int_{\Omega} C \, \varepsilon(u) : \varepsilon(v) \; dx$

Source: Y. Achdou, F. Nataf

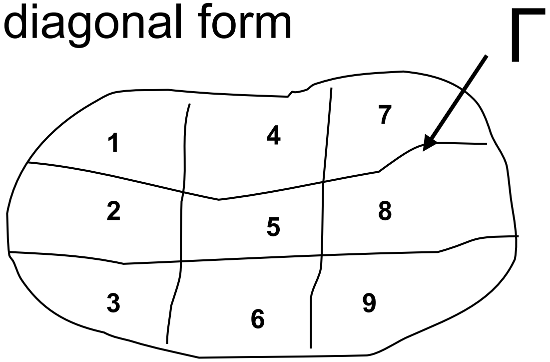


- Lack of robustness for many existing preconditioners
  - wrt jumps in coefficients / partitioning into irregular subdomains,
    e.g. one level DDM methods (Additive Schwarz, RAS), incomplete LU
  - A few small eigenvalues hinder the convergence of iterative methods

Direct factorization of a matrix in arrow block diagonal form

- Order the matrix by using k-way partitioning with vertex separators
- The permuted matrix has an arrow block diagonal form

$$A = \begin{pmatrix} A_{11} & & & A_{\Gamma 1}^T \\ & \ddots & & \vdots \\ & & A_{NN} & A_{\Gamma N}^T \\ A_{\Gamma 1} & \cdots & A_{\Gamma N} & A_{\Gamma\Gamma} \end{pmatrix}$$



- A direct factorization of A is written as

$$A = (L + D)D^{-1}(D + L^T)$$

$$= \begin{pmatrix} A_{11} & & & \\ & \ddots & & \\ & & A_{NN} & \\ A_{\Gamma 1} & \cdots & A_{\Gamma N} & S \end{pmatrix} \cdot \begin{pmatrix} A_{11}^{-1} & & & \\ & \ddots & & \\ & & A_{NN}^{-1} & \\ & & & S^{-1} \end{pmatrix} \cdot \begin{pmatrix} A_{11} & & & A_{\Gamma 1}^T \\ & \ddots & & \vdots \\ & & A_{NN} & A_{\Gamma N}^T \\ & & & S \end{pmatrix}$$

$$S = A_{\Gamma\Gamma} - \sum_{i=1}^{N} A_{\Gamma i} A_{ii}^{-1} A_{\Gamma i}^T$$

LORASC: LOw Rank Approximation based Schur Complement preconditioner

- Given A is SPD, preconditioner M is defined as

$$M = (L + D)D^{-1}(D + L^T)$$

$$= \begin{pmatrix} A_{11} & & & \\ & \ddots & & \\ & & A_{NN} & \\ A_{\Gamma 1} & \cdots & A_{\Gamma N} & \tilde{S} \end{pmatrix} \cdot \begin{pmatrix} A_{11}^{-1} & & & \\ & \ddots & & \\ & & A_{NN}^{-1} & \\ & & & \tilde{S}^{-1} \end{pmatrix} \cdot \begin{pmatrix} A_{11} & & & A_{\Gamma 1}^T \\ & \ddots & & \vdots \\ & & A_{NN} & A_{\Gamma N}^T \\ & & & \tilde{S} \end{pmatrix}$$

$$\tilde{S} \text{ approximates } S = A_{\Gamma\Gamma} - \sum_{i=1}^{N} A_{\Gamma i} A_{ii}^{-1} A_{\Gamma i}^T$$

$$\Lambda(M^{-1}A) = \Lambda(\tilde{S}^{-1}S) \cup \{1\}, \text{ where } \Lambda(M^{-1}A) = \{\lambda_{\min} = \lambda_1, \ldots, \lambda_{\max} = \lambda_n\}$$

- The approximation of *S* aims at coupling all subdomains and correcting for small eigenvalues
- E.g. the kernel of elasticity is spanned by rigid body modes, which should be included in this approximation

Approximation of the Schur complement

- We have that $\lambda_{max}(A_{\Gamma\Gamma}^{-1} S) \leq 1$
- Consider the generalized eigenvalue problem

  $Su = \lambda A_{\Gamma\Gamma} u$

  let $\lambda_{min}, \ldots, \lambda_k \leq \varepsilon$, and let $u_1, \ldots, u_k$ be the associated eigenvectors

- The Schur complement S is approximated by :

$$\tilde{S}^{-1} = A_{\Gamma\Gamma}^{-1} + U\Sigma U^T, \text{ where}$$

$$U = (u_1, \ldots, u_k), \quad \Sigma = diag(\sigma_1, \ldots, \sigma_k)$$

$$\sigma_i = \frac{\varepsilon - \lambda_i}{\lambda_i}, \quad i = 1, \ldots, k$$

- The eigenvalues of $M^{-1} A$ have values between 1 and ε

$$\varepsilon \leq \lambda(\tilde{S}^{-1} S) \leq 1$$

Results for domain decomposition methods

- AS-1: additive Schwarz

$$M_{AS-1}^{-1} = \sum_{i=1}^{N} R_i^T A_i^{-1} R_i$$

- AS-ZEM : additive Schwarz with Nicolaides like coarse space correction

$$M_{AS-ZEM}^{-1} = R_0^T \left(R_0 A R_0^T \right)^{-1} R_0 + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i,$$

where R_0 is formed by rigid body motions split by using a partition of unity

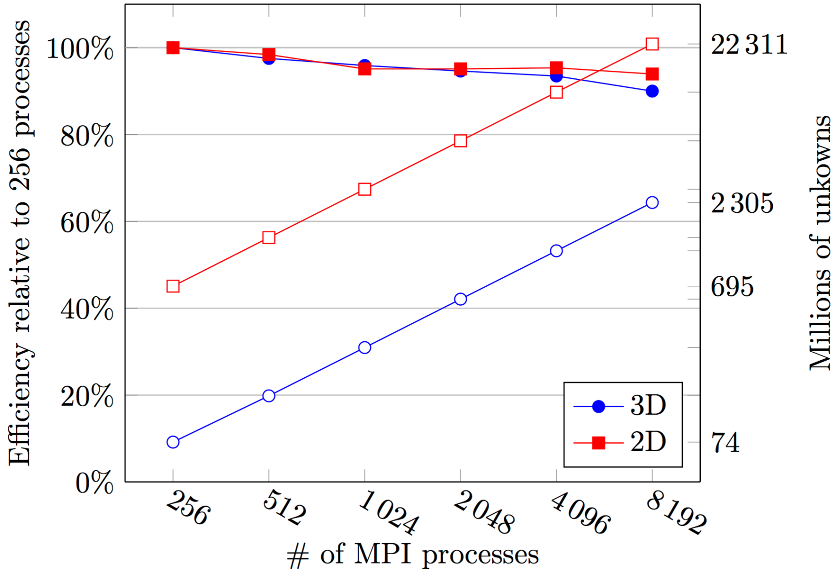
- Geneo: a recent robust two level Schwarz method [Jolivet, Nataf, Spillane et al]
  - proof of convergence of GenEO under several technical assumptions fulfilled by standard FE and bilinear forms, SPD input matrix

subd	n	AS-1	AS-ZEM (V_H)	GenEO (V_H)
4	1452	79	54 (24)	16 (46)
8	29040	177	87 (48)	16 (102)
16	58080	378	145 (96)	16 (214)

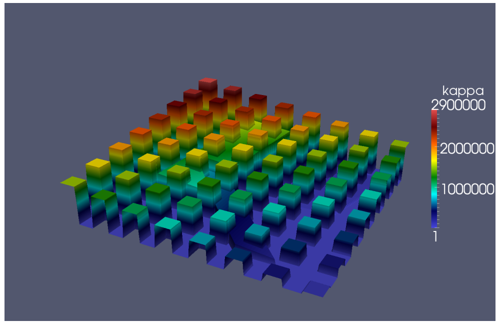
3D linear elasticity
AS-ZEM (Rigid body motions): $m_j = 6$
V_H: size of the coarse space
Results provided by F. Nataf

GenEO: weak scalability

- Darcy problem with highly heterogeneous coefficients.
- Efficiency for a 3D problem (P2 FE) – 2.3 billion unknowns – 210 secs
  and a 2D problem (P3 FE) – 22.3 billion unknowns – 180 secs



CURIE@TGCC, 92k cores, 1.6 Pflops/s,
8-cores Intel Nehalem-EX



- Remarks:
  - FreeFEM++ and MPI implementation
  - Implementation requires element stiffness matrices + connectivity

Jolivet, Hecht, Nataf, Prud'homme, Supercomputing 2013
Jolivet, Tournier, Supercomputing 2016

LORASC: convergence results

- Results for a 3D linear elasticity problem
  - CG from matlab, tolerance 10^{-8}
  - N_{mult} - number of matrix-vector operations in ARPACK
  - n_{EV} – number of deflated eigenvalues, smaller than ε

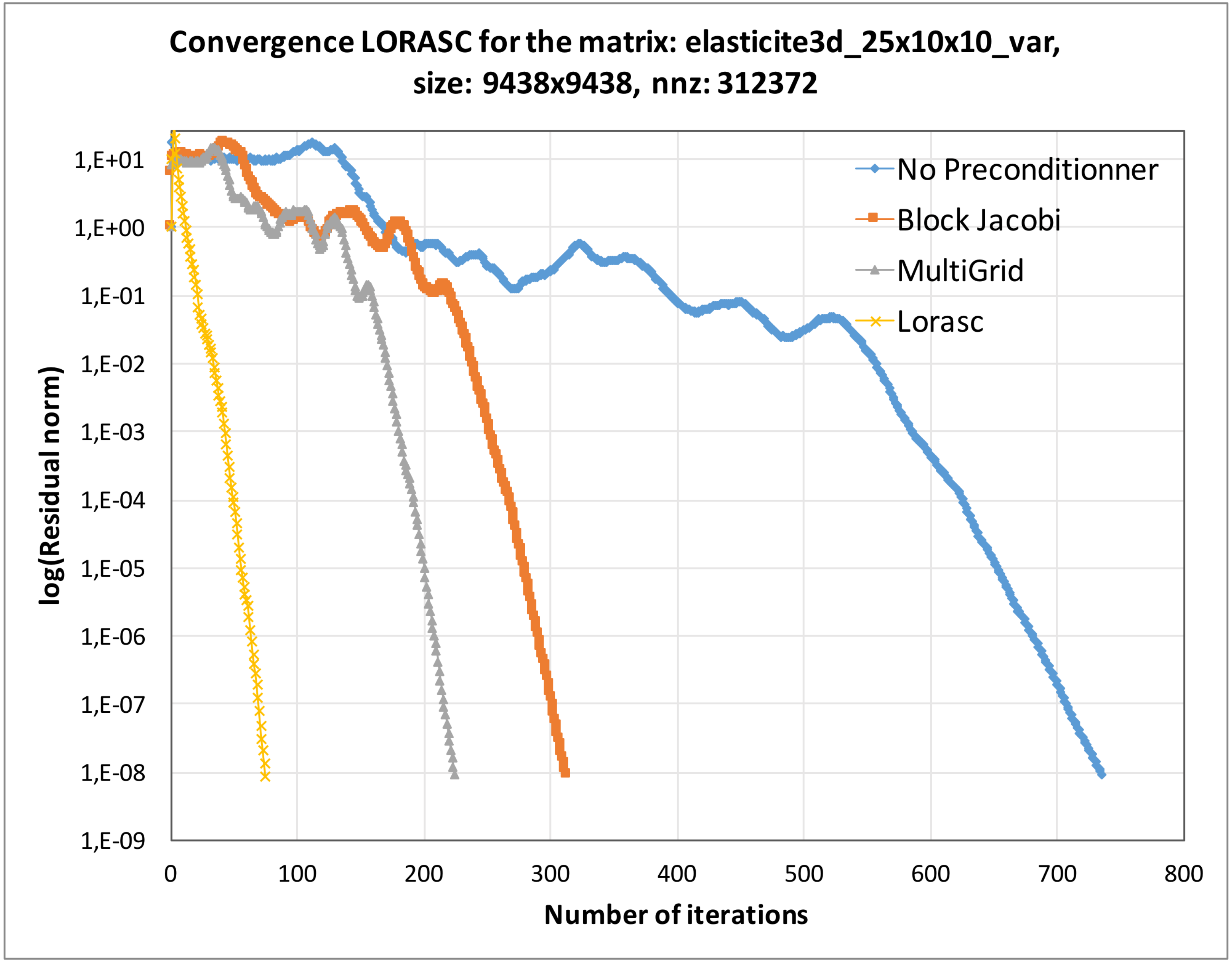
n	N_p	nnz	$\tilde{S}^{-1}, \varepsilon = 0.01$			$\tilde{S}^{-1}, \varepsilon = 0.005$			$A_{\Gamma,\Gamma}^{-1}$
			n_{EV}	N_{mult}	iter$_{\tilde{S}^{-1}}$	n_{EV}	N_{mult}	iter$_{\tilde{S}^{-1}}$	iter$_{A_{\Gamma,\Gamma}^{-1}}$
4719	2	153057	0	0	71	0	0	71	71
9438	4	312372	5	92	65	3	83	89	113
18513	8	618747	10	111	63	8	95	84	207
36663	16	1231497	15	132	60	11	111	76	267
72963	32	2456997	42	325	55	24	230	64	592

Weak scaling results

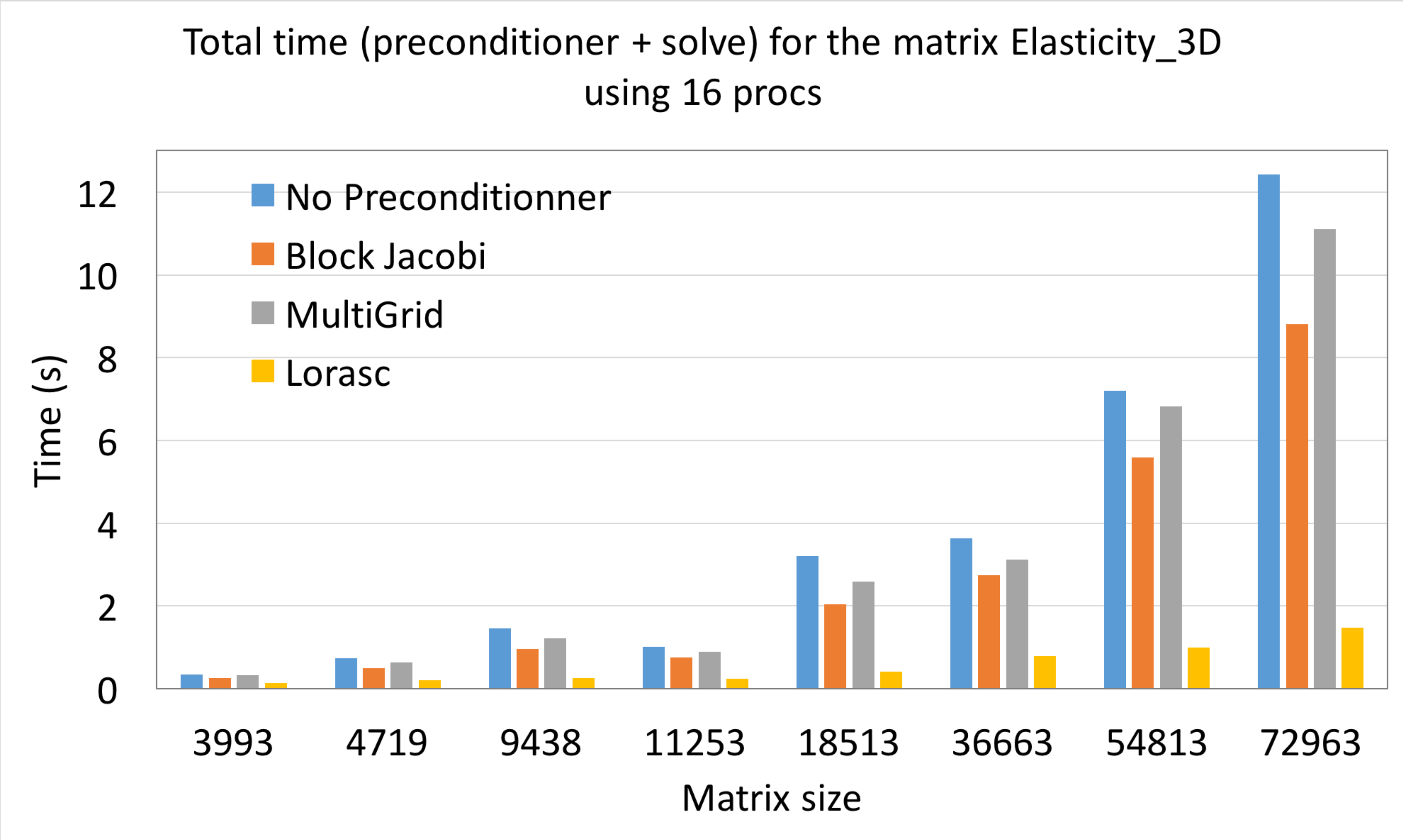
n	N_p	nnz	$\tilde{S}^{-1}, \varepsilon = 0.01$			$\tilde{S}^{-1}, \varepsilon = 0.005$			$A_{\Gamma,\Gamma}^{-1}$
			n_{EV}	N_{mult}	iter$_{\tilde{S}^{-1}}$	n_{EV}	N_{mult}	iter$_{\tilde{S}^{-1}}$	iter$_{A_{\Gamma,\Gamma}^{-1}}$
72963	2	2456997	6	83	58	4	83	74	87
72963	4	2456997	13	119	65	8	110	75	168
72963	8	2456997	23	202	61	13	119	74	322
72963	16	2456997	32	249	61	18	159	69	465
72963	32	2456997	42	325	55	24	230	64	592

Strong scaling results

Convergence of Lorasc



Performance on small number of processors



Conclusions

- Need to redesign/reformulate algorithms to reduce communication
  - Derive when possible lower bounds on communication
  - Minimize communication at the cost of redundant computation
  - Communication avoiding algorithms often faster than conventional algorithms in practice

- Remains a lot to do for sparse linear algebra
  - Communication bounds, communication optimal algorithms
  - Preconditioners - limited by memory and communication, not flops

- And BEYOND

Collaborators, funding

Collaborators:

- Inria Alpines: A. Ayala, S. Cayrols, S. Donfack, A. Khabou, M. Jacquelin, S. Moufawad, F. Nataf, CNRS, O. Tissot, H. Xiang, S. Yousef
- J. Demmel (UC Berkeley), G. Ballard (Sandia), B. Gropp (UIUC), M. Gu (UC Berkeley), M. Hoemmen (Sandia), N. Knight (NYU), J. Langou (CU Denver), V. Kale (UIUC), P. Henon (Total), P. Ricoux (Total)

Thanks to: EC H2020 NLAFET, Total, ANR Petal and Petalh projects, ANR Midas, Digiteo Xscale NL, COALA INRIA funding

Further information:

http://www-rocq.inria.fr/who/Laura.Grigori/

References

Results presented from:

- J. Demmel, L. Grigori, M. F. Hoemmen, and J. Langou, *Communication-optimal parallel and sequential QR and LU factorizations*, UCB-EECS-2008-89, 2008, SIAM journal on Scientific Computing, Vol. 34, No 1, 2012.
- L. Grigori, J. Demmel, and H. Xiang, *Communication avoiding Gaussian elimination*, Proceedings of the IEEE/ACM SuperComputing SC08 Conference, November 2008.
- L. Grigori, J. Demmel, and H. Xiang, *CALU: a communication optimal LU factorization algorithm*, SIAM. J. Matrix Anal. & Appl., 32, pp. 1317-1350, 2011.
- L. Grigori, P.-Y. David, J. Demmel, and S. Peyronnet, *Brief announcement: Lower bounds on communication for sparse Cholesky factorization of a model problem*, ACM SPAA 2010.
- S. Donfack, L. Grigori, and A. Kumar Gupta, *Adapting communication-avoiding LU and QR factorizations to multicore architectures*, Proceedings of IEEE International Parallel & Distributed Processing Symposium IPDPS, April 2010.
- S. Donfack, L. Grigori, W. Gropp, and V. Kale, *Hybrid static/dynamic scheduling for already optimized dense matrix factorization* , Proceedings of IEEE International Parallel & Distributed Processing Symposium IPDPS, 2012.
- J. Demmel, L. Grigori, M. Gu, H. Xiang, Communication avoiding rank revealing QR factorization with column pivoting, SIAM J. Matrix Anal. & Appl, Vol. 36, No. 1, pp. 55-89, 2015.
- G. Ballard, J. Demmel, L. Grigori, M. Jacquelin, N. Knight, J. D. Nguyen, and E. Solomonik *Reconstructing Householder vectors for QR factorization*, Journal of Parallel and Distributed Computing, in press, 2015.
- L. Grigori, S. Moufawad, F. Nataf, *Enlarged Krylov Subspace Conjugate Gradient Methods for Reducing Communication* , INRIA TR 8597.
- H. Al Daas, L. Grigori, P. Henon, P. Ricoux, Enlarged GMRES, In preparation.