Avoiding communication in linear algebra

Laura Grigori
ALPINES
INRIA Rocquencourt - LJLL, UPMC

- SIAM Conference on Parallel Processing – Spring 2016
    - Organized by SIAG on Supercomputing
    - Very likely to be organized in Paris

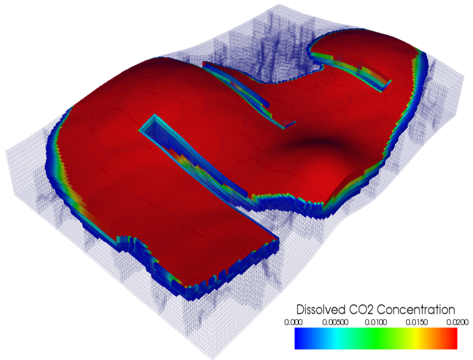
Plan

- Motivation
- Selected past work on reducing communication
- Communication complexity of linear algebra operations
- Communication avoiding for dense linear algebra
  - LU, QR, Rank Revealing QR factorizations
  - Progressively implemented in ScaLAPACK or LAPACK
  - Algorithms for multicore processors
- Communication avoiding for sparse linear algebra
  - Iterative methods and preconditioning
- Conclusions

Data driven science

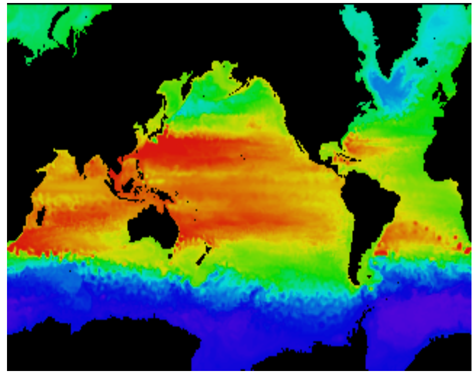
Numerical simulations require increasingly computing power as data sets grow exponentially

CO2 Underground storage



Source: T. Guignon, IFPEN

Climate modeling



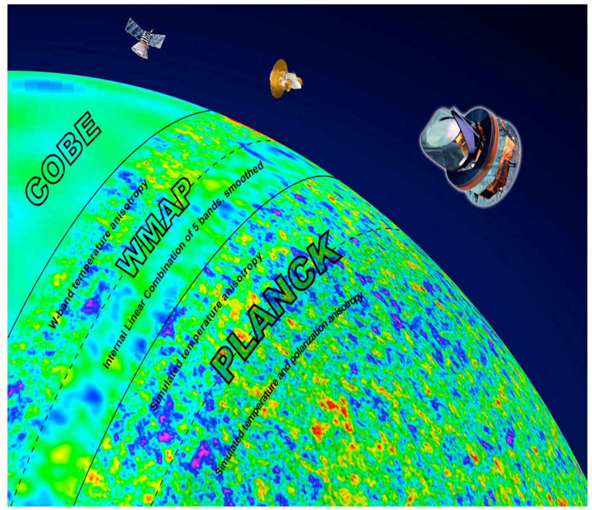
http://www.epm.ornl.gov/chammp/chammp.html

Figures from astrophysics:

- Produce and analyze multi-frequency 2D images of the universe when it was 5% of its current age.
- COBE (1989) collected 10 gigabytes of data, required 1 Teraflop per image analysis.
- PLANCK (2010) produced 1 terabyte of data, requires 100 Petaflops per image analysis.
- CMBPol (2020) is estimated to collect .5 petabytes of data, will require 100 Exaflops per image analysis.

Source: J. Borrill, LBNL, R. Stompor, Paris 7

Astrophysics: CMB data analysis



http://www.scidacreview.org/0704/html/cmb.html

Motivation - the communication wall

- Runtime of an algorithm is the sum of:
  - #flops x **time_per_flop**
  - #words_moved / **bandwidth**
  - #messages x **latency**

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

Annual improvements			
Time/flop		Bandwidth	Latency
59%	Network	**26%**	**15%**
	DRAM	**23%**	**5%**

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

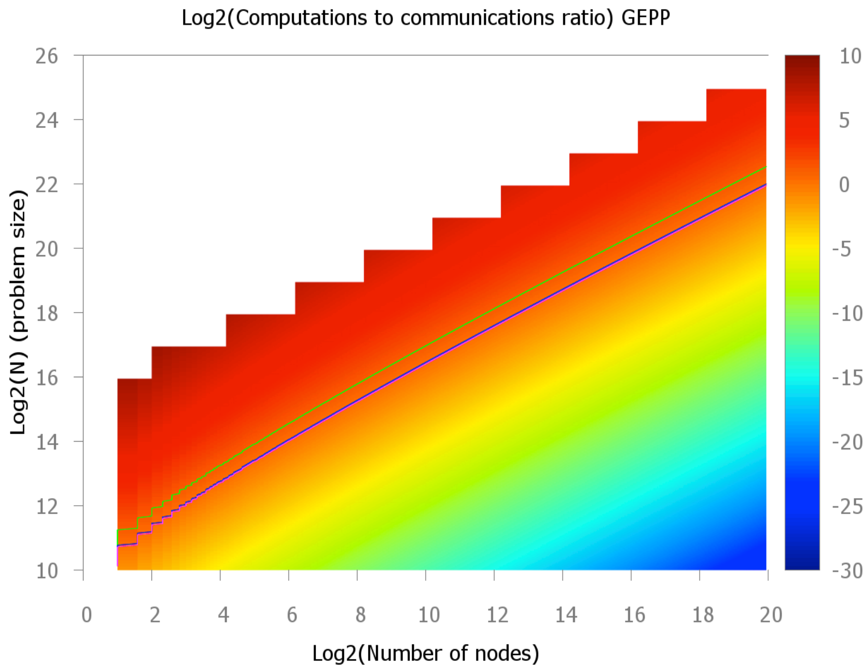
*"We are going to hit the **memory wall**, unless something basic changes"* [W. Wulf, S. McKee, 95]

Motivation

- The communication problem needs to be taken into account higher in the computing stack
- A paradigm shift in the way the numerical algorithms are devised is required
- Communication avoiding algorithms - a novel perspective for numerical linear algebra
  - Minimize volume of communication
  - Minimize number of messages
  - Minimize over multiple levels of memory/parallelism
  - Allow redundant computations (preferably as a low order term)

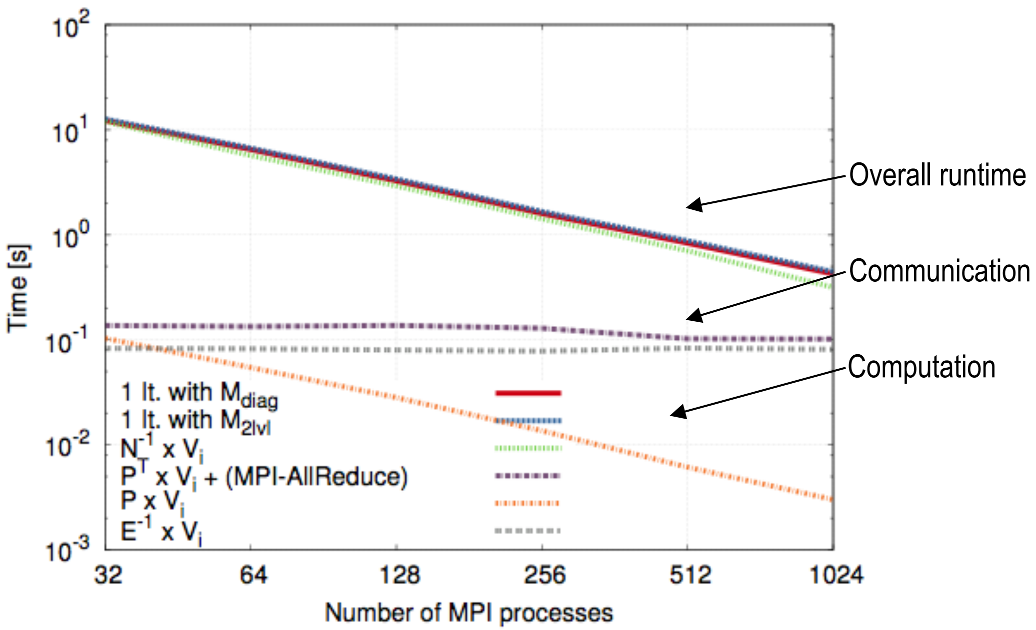
Previous work on reducing communication

- Tuning
  - Overlap communication and computation, at most a factor of 2 speedup
- Ghosting
  - Store redundantly data from neighboring processors for future computations
- Scheduling
  - Block algorithms for linear algebra
    - Barron and Swinnerton-Dyer, 1960
    - ScaLAPACK, Blackford et al 97
  - Cache oblivious algorithms for linear algebra
    - Gustavson 97, Toledo 97, Frens and Wise 03, Ahmed and Pingali 00

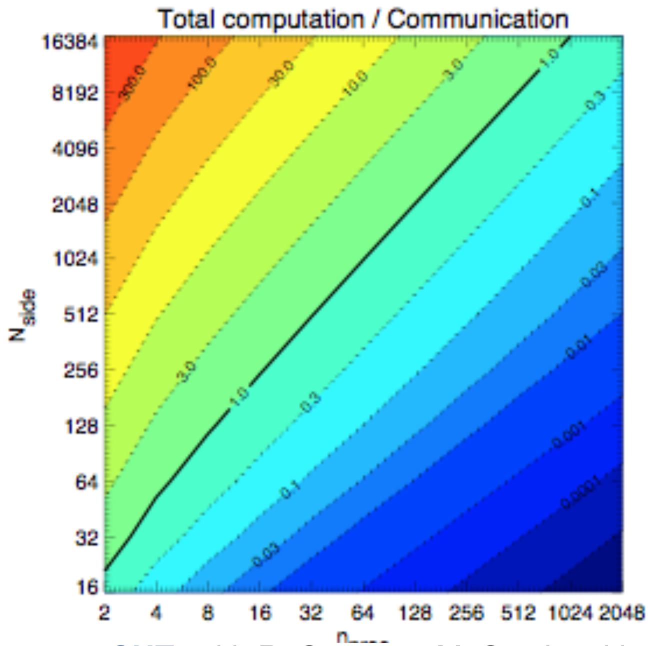


Communication in CMB data analysis

- Map-making problem
  - Find the best map *x* from observations *d*, scanning strategy *A*, and noise N^{-1}
  - Solve generalized least squares problem involving sparse matrices of size 10^{12}-by-10^{7}
- Spherical harmonic transform (SHT)
  - Synthesize a sky image from its harmonic representation
    - Computation over rows of a 2D object (summation of spherical harmonics)
    - Communication to transpose the 2D object
    - Computation over columns of the 2D object (FFTs)



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



SHT, with R. Stompor, M. Szydlarski
Simulation on a petascale computer

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 become
  #words_moved $\geq \Omega (n^2 / P^{1/2})$, #messages $\geq \Omega (P^{1/2})$

Algorithm	Minimizing #words (not #messages)	Minimizing #words and #messages
Cholesky	ScaLAPACK	ScaLAPACK
LU	ScaLAPACK uses 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	[Branescu, Demmel, LG, Gu, Xiang 11] uses tournament pivoting, 3x flops

- Only several references shown, block algorithms (ScaLAPACK) and communication avoiding algorithms

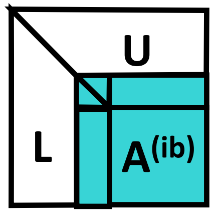
LU factorization (as in ScaLAPACK pdgetrf)

LU factorization on a P = P_r x P_c grid of processors

For ib = 1 to n-1 step b

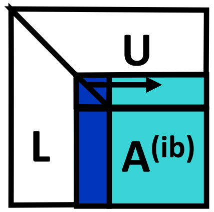
$A^{(ib)}$ = A(ib:n, ib:n)

#messages



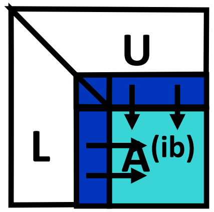
(1) Compute panel factorization — $O(n \log_2 P_r)$

- find pivot in each column, swap rows



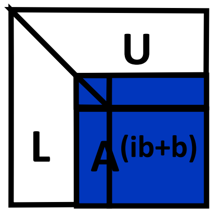
(2) Apply all row permutations — $O(n/b(\log_2 P_c + \log_2 P_r))$

- broadcast pivot information along the rows
- swap rows at left and right



(3) Compute block row of U — $O(n/b \log_2 P_c)$

- broadcast right diagonal block of L of current panel



(4) Update trailing matrix — $O(n/b(\log_2 P_c + \log_2 P_r))$

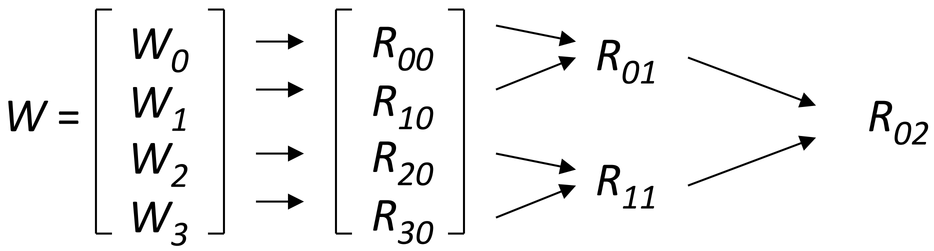
- broadcast right block column of L
- broadcast down block row of U

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

- QR decomposition of m x b matrix W, m >> b
  - P processors, block row layout

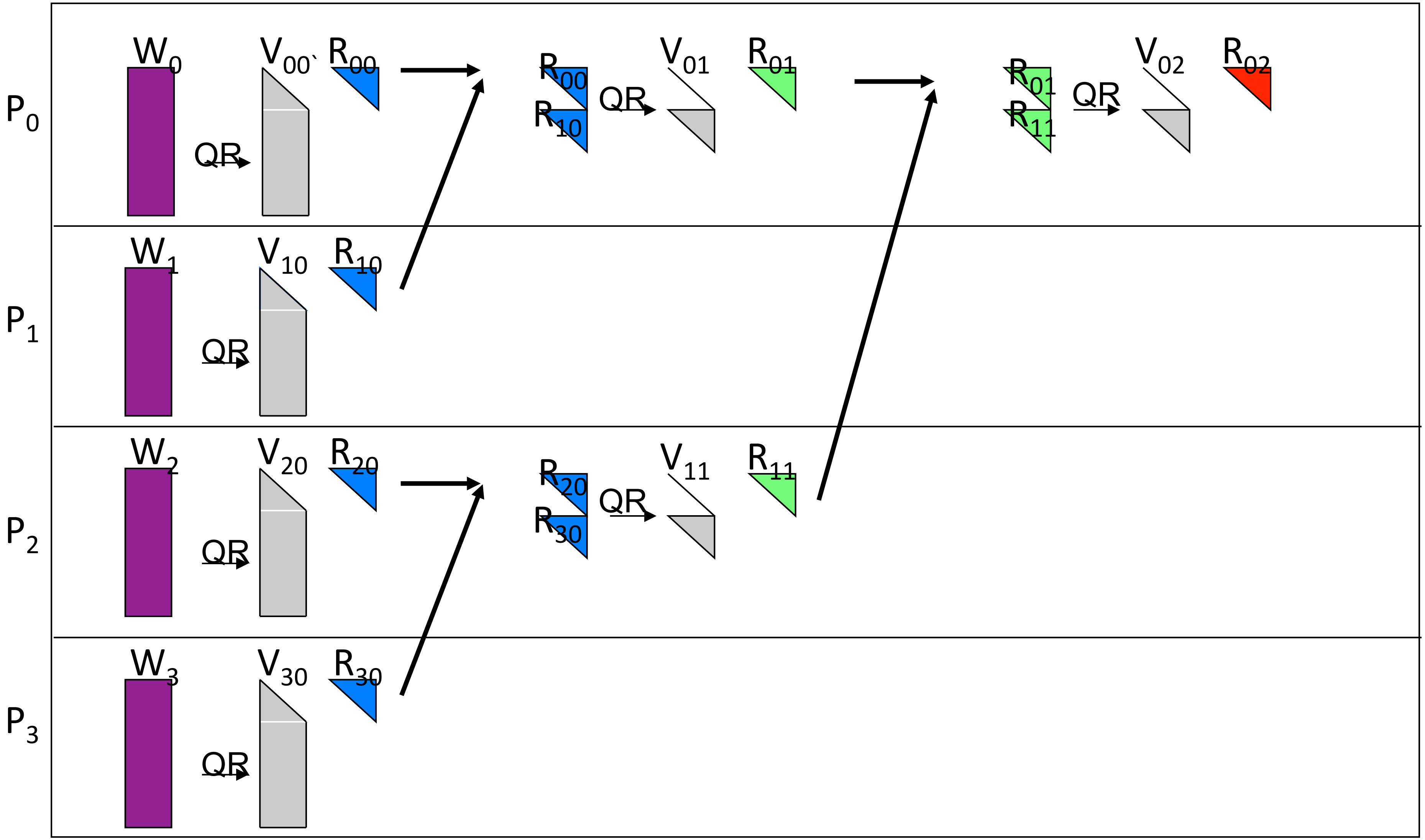
- Classic Parallel Algorithm
  - Compute Householder vector for each column
  - Number of messages ∝ b log P

- Communication Avoiding Algorithm
  - Reduction operation, with QR as operator
  - Number of messages ∝ log P



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

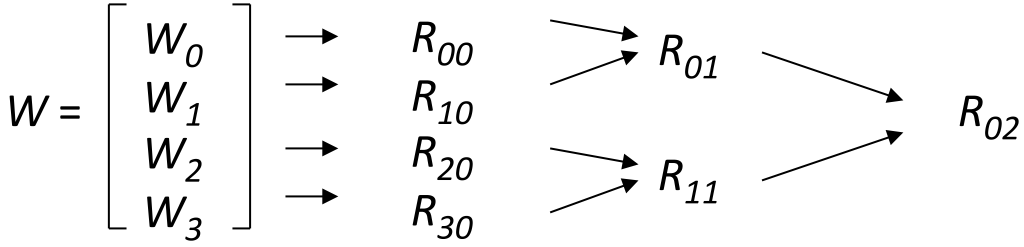
Parallel TSQR



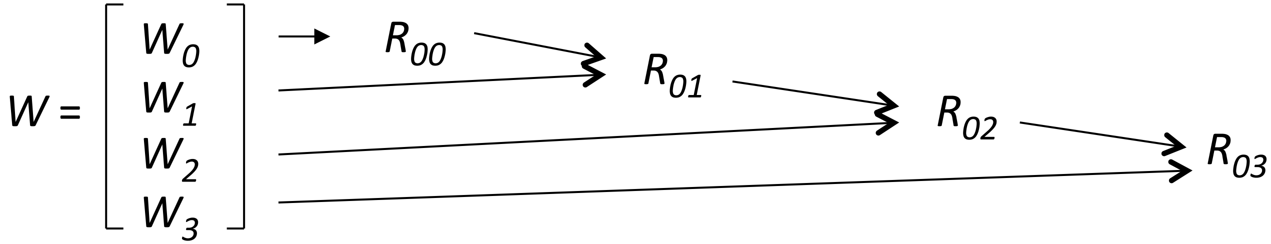
References: Golub, Plemmons, Sameh 88, Pothen, Raghavan, 89, Da Cunha, Becker, Patterson, 02

Flexibility of TSQR and CAQR algorithms

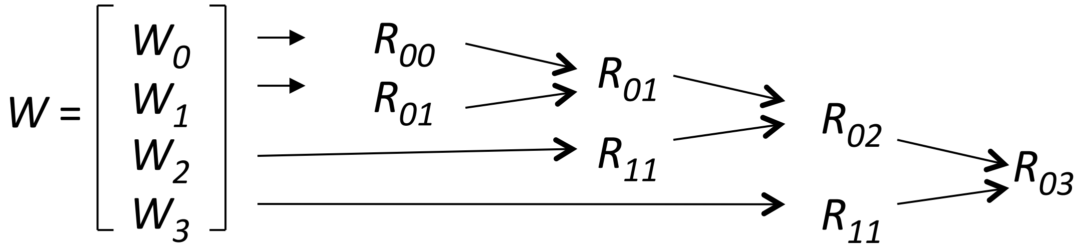
Parallel:



Sequential:

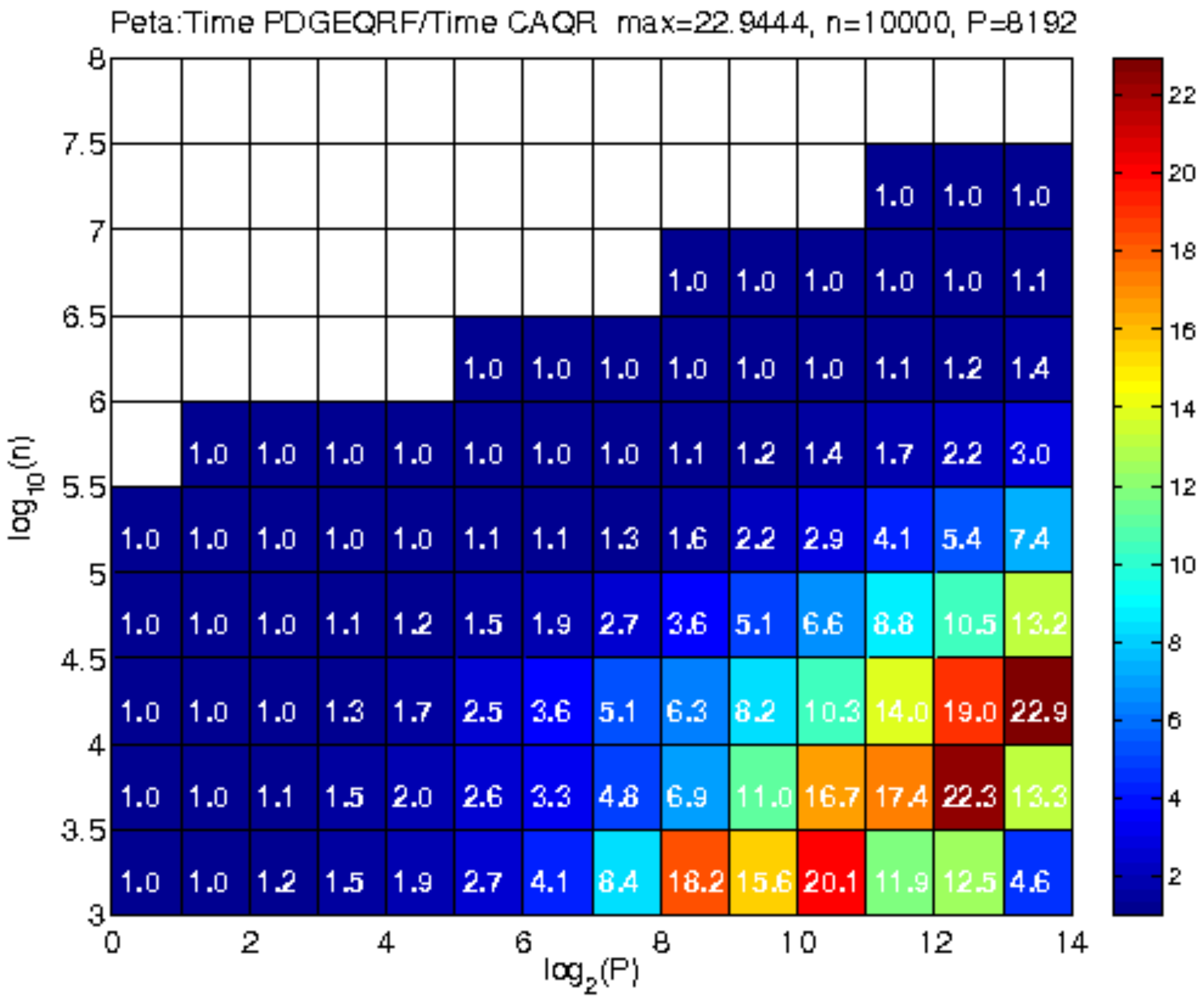


Dual Core:



Reduction tree will depend on the underlying architecture, could be chosen dynamically

Modeled Speedups of CAQR vs ScaLAPACK



Petascale
up to 22.9x

IBM Power 5
up to 9.7x

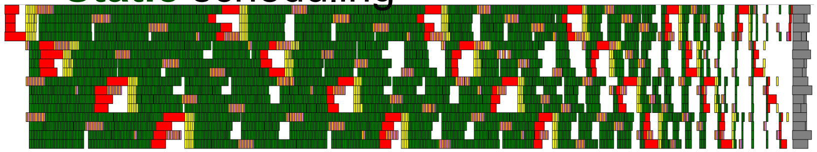
“Grid”
up to 11x

Petascale machine with 8192 procs, each at 500 GFlops/s, a bandwidth of 4 GB/s.

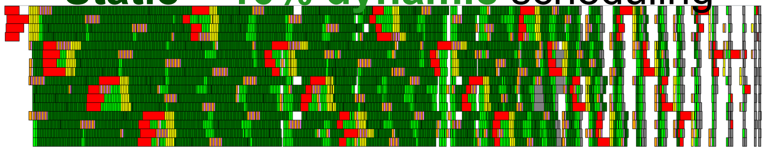
$$\gamma = 2\cdot 10^{-12} s, \alpha = 10^{-5} s, \beta = 2\cdot 10^{-9} s / word.$$

Lightweight scheduling for CALU

Static scheduling

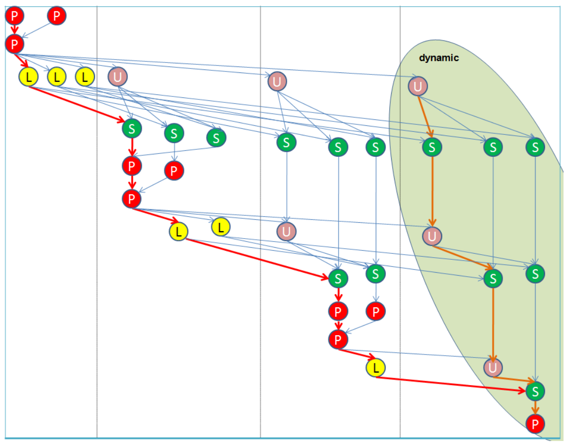


Static + **10% dynamic** scheduling



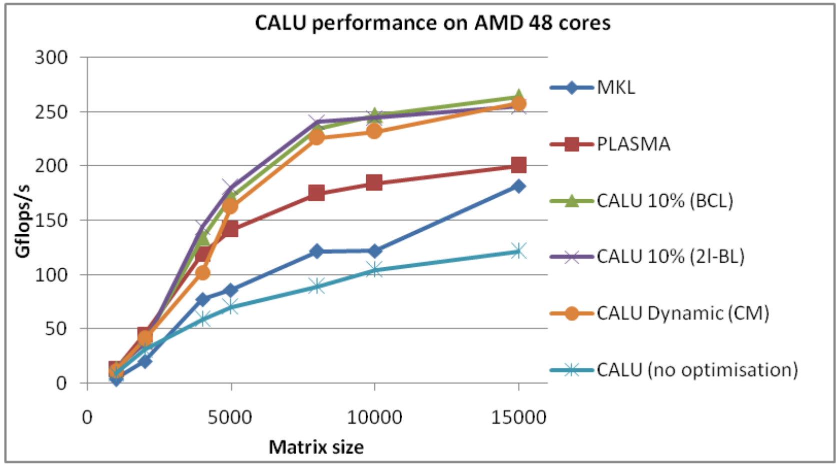
100% dynamic scheduling

time



Task dependency graph of CALU

Donfack, LG, Gropp, Kale, IPDPS 2012



Plan

- Motivation
- Selected past work on reducing communication
- Communication complexity of linear algebra operations
- Communication avoiding for dense linear algebra
  - LU, LU_PRRP, QR, Rank Revealing QR factorizations
  - Often not in ScaLAPACK or LAPACK
  - Algorithms for multicore processors
- Communication avoiding for sparse linear algebra
  - Iterative methods and preconditioning
- Conclusions

Preconditioned Krylov subspace methods

- Solve $Ax=b$ by using iterative methods

  Find 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.

- Convergence depends on $\kappa(A)$ and the eigenvalue distribution (for SPD matrices).

- To accelerate convergence, solve $M^{-1}Ax = M^{-1}b$

- SAGE preconditioner – with F. Nataf and S. Yousef
  - Fully algebraic robust preconditioner
  - Based on solving a generalized eigenvalue problem

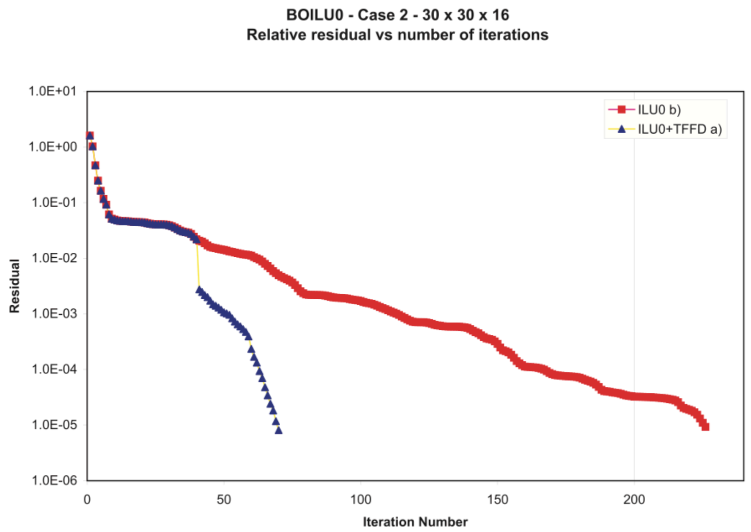
Challenge in getting scalable preconditioners

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

Approaches to deal with low frequency modes

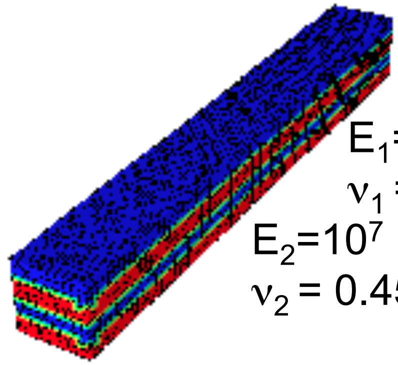
- Deflation through augmentation or preconditioning
- Two level domain decomposition methods, e.g.:
  - Geneo: a robust two level Schwarz method [Jolivet, Nataf, Spillane et al]
  - Based on solving local generalized eigenvalue problems
  - Requires information from the underlying PDE.

- Direction preserving preconditioners $MT = AT$
  - Filtering factorization, Wagner, Wittum (1997), Achdou, Nataf (2001)
  - Direction preserving semiseparable approximation of SPD matrices, Gu, Li, Vassilevski (2010)
    - If the near null-space of the original fine grid matrix is preserved, then view the preconditioner as a coarse discretization matrix
  - Multigrid methods
    - Bootstrup AMG (Brandt, Brannick, Kahl, and Livshits)

Numerical results

- Linear elasticity problems

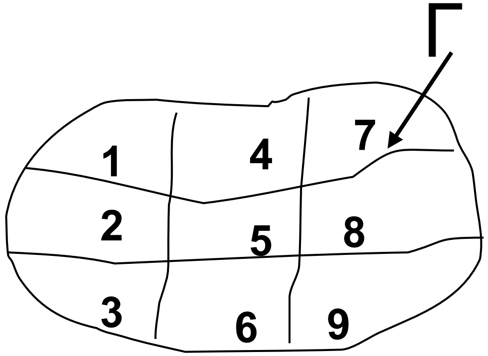
- Results obtained by using domain decomposition methods
  - AS-1: additive Schwarz
  - AS-ZEM : additive Schwarz with Nicolaides coarse space correction
  - Geneo: a recent robust two level Schwarz method [Jolivet, Nataf, Spillane et al]
    - proof of convergence of GenEO under several technical assumptions fulfilled by standard FE and bilinear forms, SPD input matrix



subd	dofs	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)

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

SAGE: Schur complement Approximation based on a Generalized Eigenvalue problem



- Given A is SPD, preconditioner M is defined as

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

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

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

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

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

- The Schur complement S is approximated by :

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

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

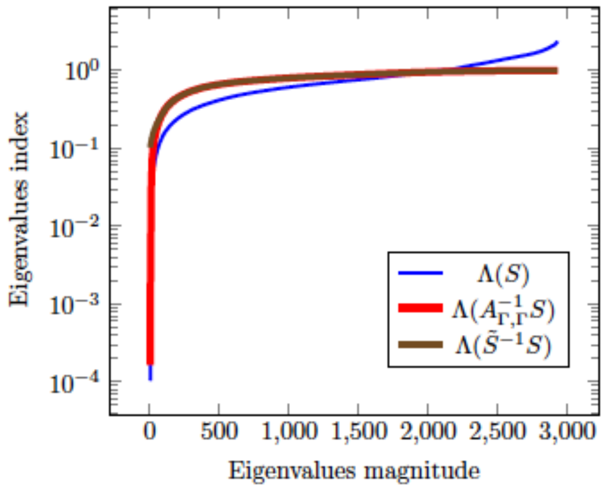
$$\sigma_i = \frac{\tau - \lambda_i}{\lambda_i}, \ \ i = 1, \ldots, k$$

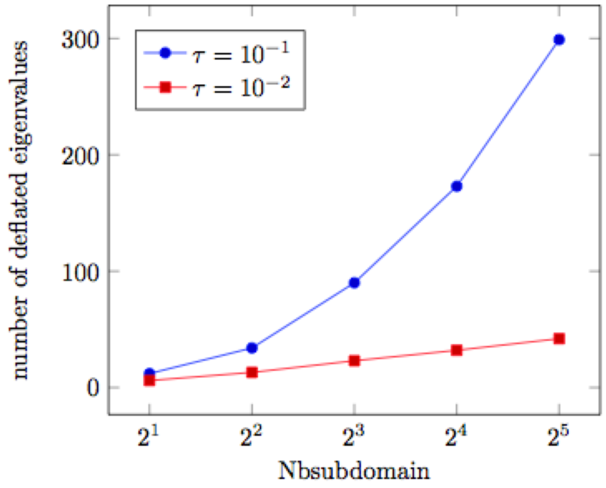
- The condition number of $M^{-1} A$ is bounded by τ^{-1} since

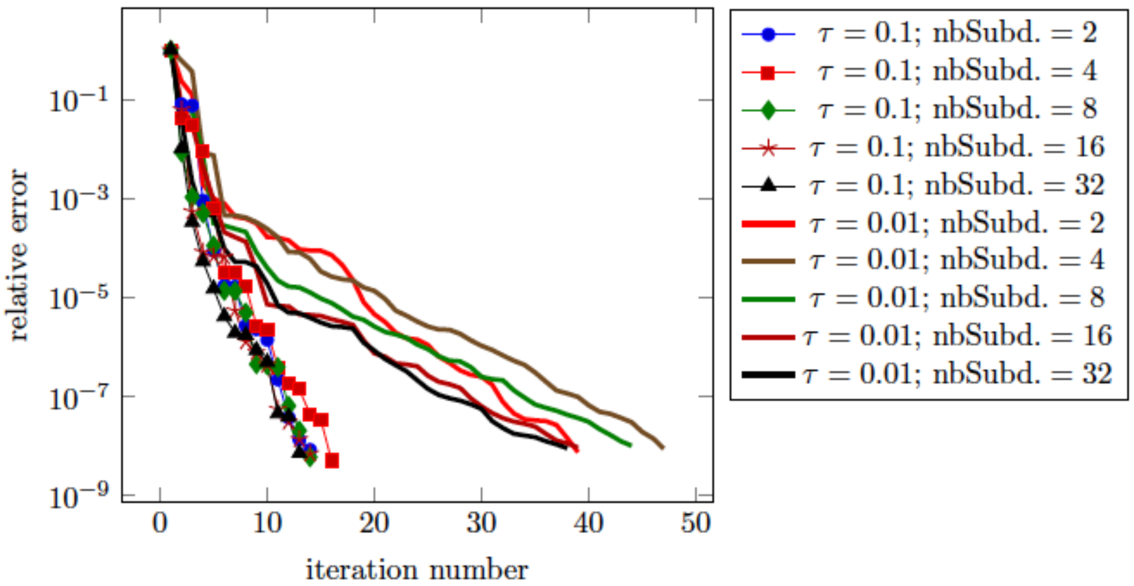
$$\tau \leq \lambda(\tilde{S}^{-1} S) \leq 1$$

SAGE: numerical results

- Results for a 3D problem, ndofs 72963, no of nonzeros 2456997







Conclusions

- Introduced a new class of communication avoiding algorithms that minimize communication
  - Attain theoretical lower bounds on communication
  - Minimize communication at the cost of redundant computation
  - Are often faster than conventional algorithms in practice
- Remains a lot to do for sparse linear algebra
  - Communication bounds, communication optimal algorithms
  - Enlarged Krylov subspace solvers
  - Preconditioners - limited by memory and communication, not flops
- And BEYOND

Collaborators, funding

Collaborators:

- S. Donfack, INRIA, A. Khabou, INRIA, M. Jacquelin, INRIA, L. Qu, Paris 11, F. Nataf, CNRS, S. Moufawad, INRIA, S. Youssef, Inria, H. Xiang, Wuhan University
- J. Demmel, UC Berkeley, B. Gropp, UIUC, M. Gu, UC Berkeley, M. Hoemmen, UC Berkeley, J. Langou, CU Denver, V. Kale, UIUC

Funding: 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, published in 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.
- M. Hoemmen's Phd thesis, *Communication avoiding Krylov subspace methods*, 2010.
- 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.
- A. Khabou, J. Demmel, L. Grigori, and M. Gu, *LU factorization with panel rank revealing pivoting and its communication avoiding version*, LAWN 263, 2012.
- L. Grigori, S. Moufawad, *Communication avoiding incomplete LU preconditioner*, in preparation, 2012