Iterative methods, preconditioning, and their application to CMB data analysis

Laura Grigori
INRIA Saclay

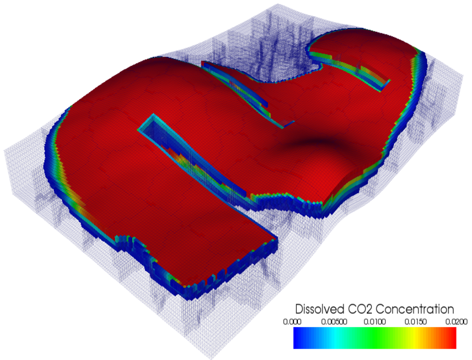
Plan

- Motivation
- Communication avoiding for numerical linear algebra
  - Novel algorithms that minimize communication
  - Often not in ScaLAPACK or LAPACK (YET !)
  - Iterative methods and preconditioning
- Application to CMB data analysis in astrophysics
- Conclusions

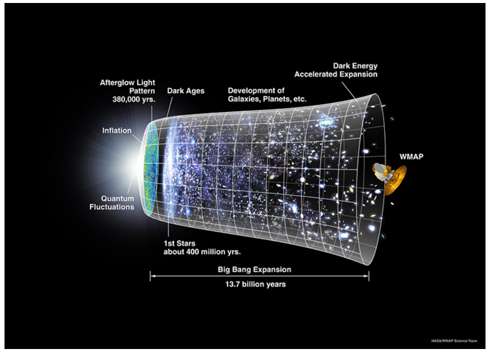
Data driven science

Numerical simulations require increasingly computing power as data sets grow exponentially

CO2 Underground storage



History of the universe

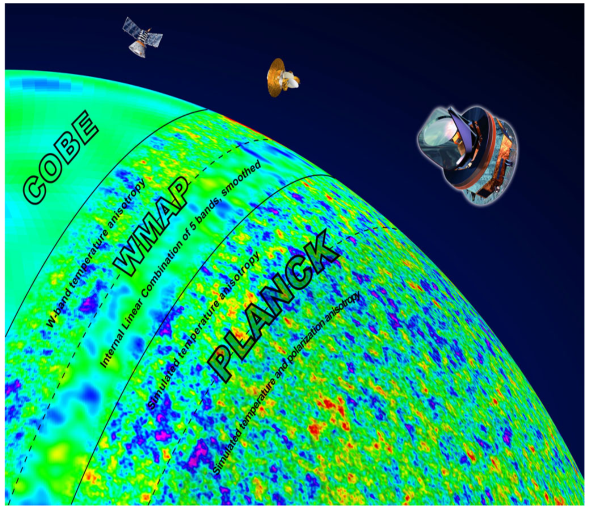


Figures from astrophysics:

- 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



The communication wall

- 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%

- Real performance << peak performance

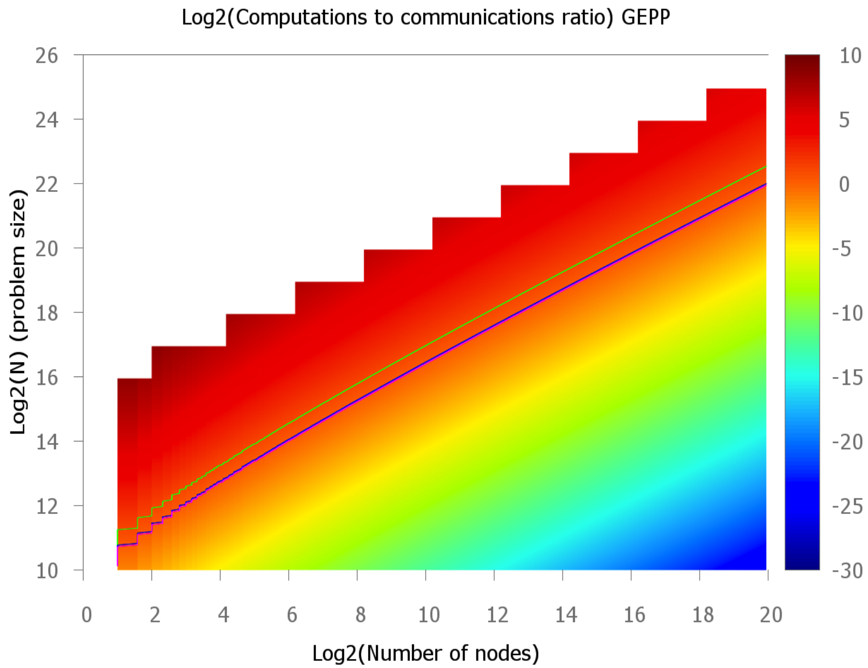
- Our goal - take the communication problem higher in the computing stack
- Communication avoiding algorithms- a novel perspective for linear algebra
  - Minimize volume of communication
  - Minimize number of messages
- Communication avoiding implies energy reduction

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

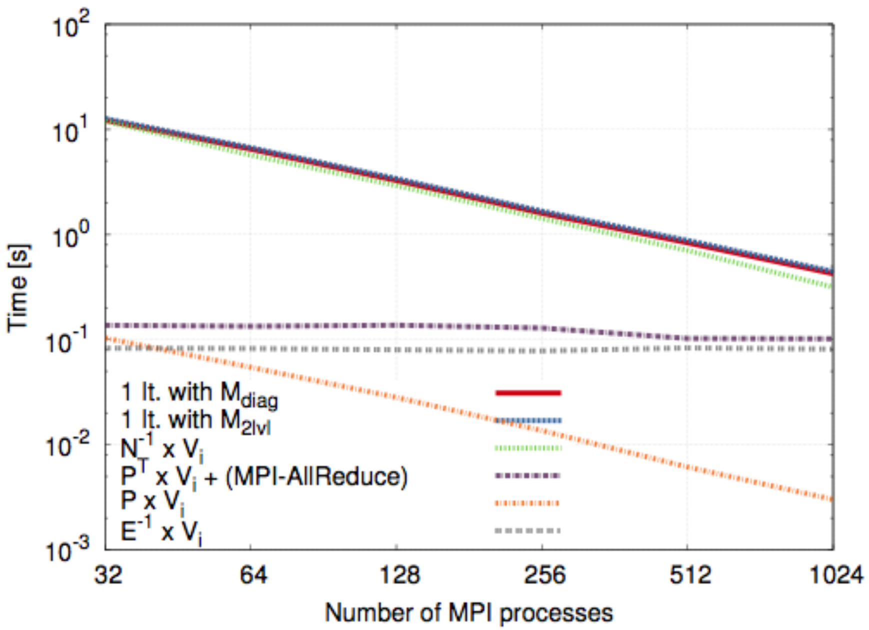
- Scheduling
  - Cache oblivious algorithms for linear algebra
    - Gustavson 97, Toledo 97, Frens and Wise 03, Ahmed and Pingali 00
  - Block algorithms for linear algebra
    - ScaLAPACK, Blackford et al 97



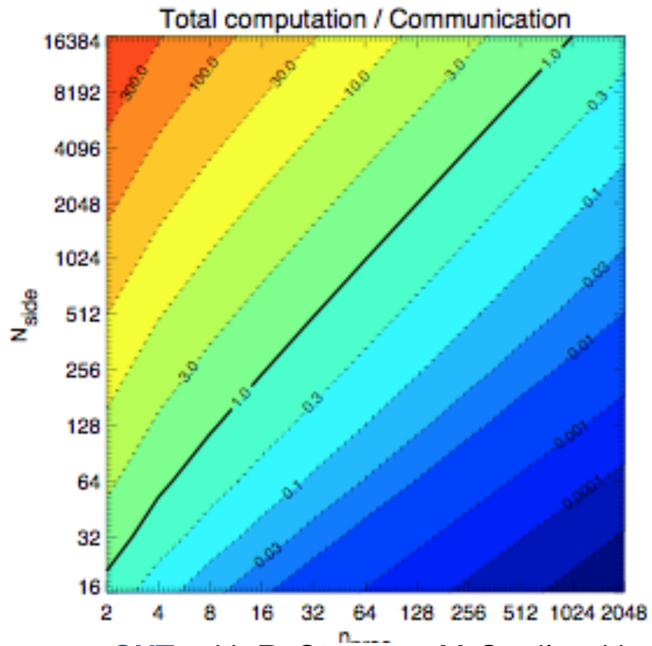
Courtesy M. Jacquelin

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

Parallel algorithms and communication bounds

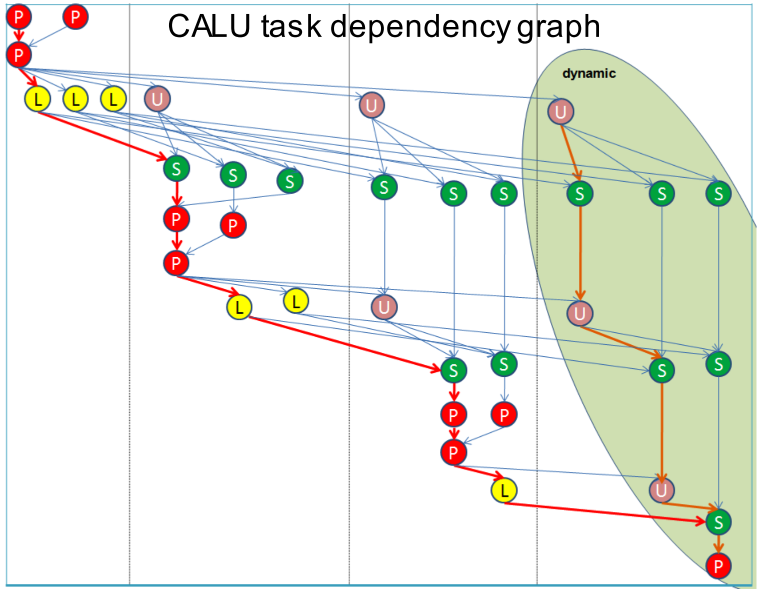
- If memory per processor = n^2 / P, the lower bounds become
  #words_moved ≥ $\Omega (n^2 / P^{1/2})$, #messages ≥ $\Omega (P^{1/2})$
  Hong and Kung, 81, Irony et al, 04, Demmel et al, 11.

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 uses column pivoting	[Branescu, Demmel, LG, Gu, Xiang 11] uses tournament pivoting, 3x flops

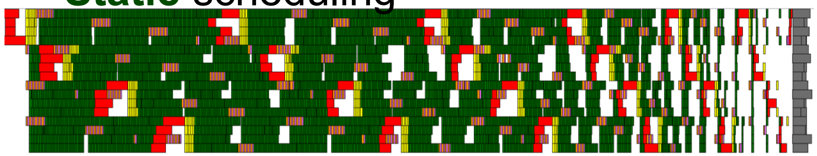
- Only several references shown, block algorithms (ScaLAPACK) and communication avoiding algorithms

Best performance of CALU on multicore architectures

- Based on lightweight scheduling - a self-adaptive strategy to provide
  - A good trade-off between load balance, data locality, and dequeue overhead.
  - Performance consistency
  - Shown to be efficient for regular mesh computation [B. Gropp and V. Kale]
  - S. Donfack, LG, B. Gropp, V. Kale, IPDPS'12

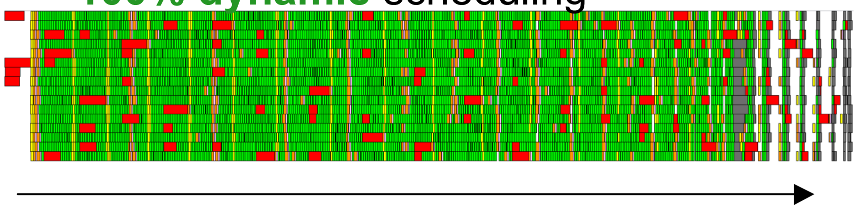


Static scheduling

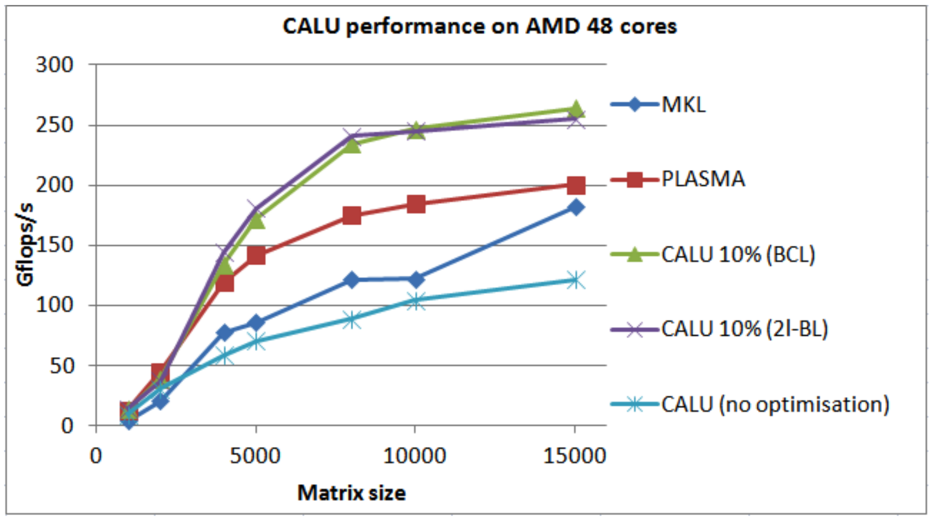


Static + **10% dynamic** scheduling

100% dynamic scheduling



time



Communication in Krylov subspace methods

Iterative methods to solve $Ax = b$

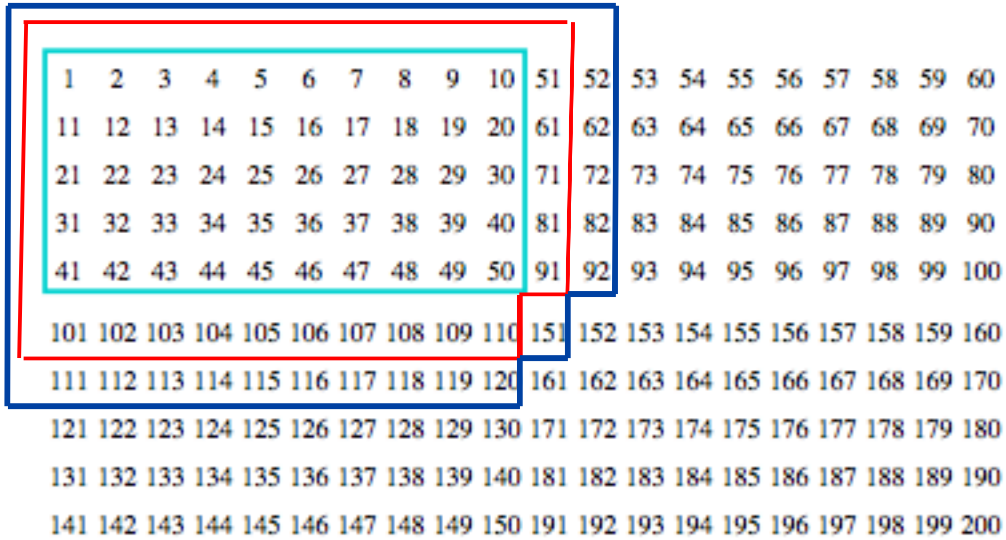
- Find a solution x_k from $x_0 + K_k(A, r_0)$, where $K_k(A, r_0) = span\{r_0, A r_0, \ldots, A^{k-1} r_0\}$ such that the Petrov-Galerkin condition $b - Ax_k \perp L_k$ is satisfied.
- For numerical stability, an orthonormal basis $\{q_1, q_2, \ldots, q_k\}$ for $K_k(A, r_0)$ is computed (CG, GMRES, BiCGstab,…)
- Each iteration requires
  - Sparse matrix vector product
  - Dot products for the orthogonalization process
- *S-step Krylov subspace methods*
  - Unroll s iterations, orthogonalize every s steps

- Van Rosendale '83, Walker '85, Chronopoulous and Gear '89, Erhel '93, Toledo '95, Bai, Hu, Reichel '91 (Newton basis), Joubert and Carey '92 (Chebyshev basis), etc.
- Recent references: G. Atenekeng, B. Philippe, E. Kamgnia (to enable multiplicative Schwarz preconditioner), J. Demmel, M. Hoemmen, M. Mohiyuddin, K. Yellick (to minimize communication, next slide)

S-step Krylov subspace methods

- To avoid communication, unroll s steps, ghost necessary data,
  - generate a set of vectors W for the Krylov subspace $K_k(A, r_0)$
  - orthogonalize the vectors using TSQR(W)

Domain and ghost data to compute $A^2 x$ with no communication →



Example: 5 point stencil 2D grid partitioned on 4 processors

- A factor of O(s) less data movement in the memory hierarchy
- A factor of O(s) less messages in parallel

Research opportunities and limitations

Length of the basis "s" is limited by

- Size of ghost data
- Loss of precision

Cost for a 3D regular grid, 7 pt stencil

s-steps	Memory	Flops
GMRES	$O(s\ n/P)$	$O(s\ n/P)$
CA-GMRES	$O(s\ n/P)+$ $O(s\ (n/P)^{2/3})+$ $O(s^2\ (n/P)^{1/3})$	$O(s\ n/P)+$ $O(s^2\ (n/P)^{2/3})+$ $O(s^3\ (n/P)^{1/3})$

Preconditioners: few identified so far to work with s-step methods

- Highly decoupled preconditioners: Diagonal, Block Jacobi
- Hierarchical, semiseparable matrices (M. Hoemmen, J. Demmel)
- Incomplete LU factorizations (LG, S. Moufawad)

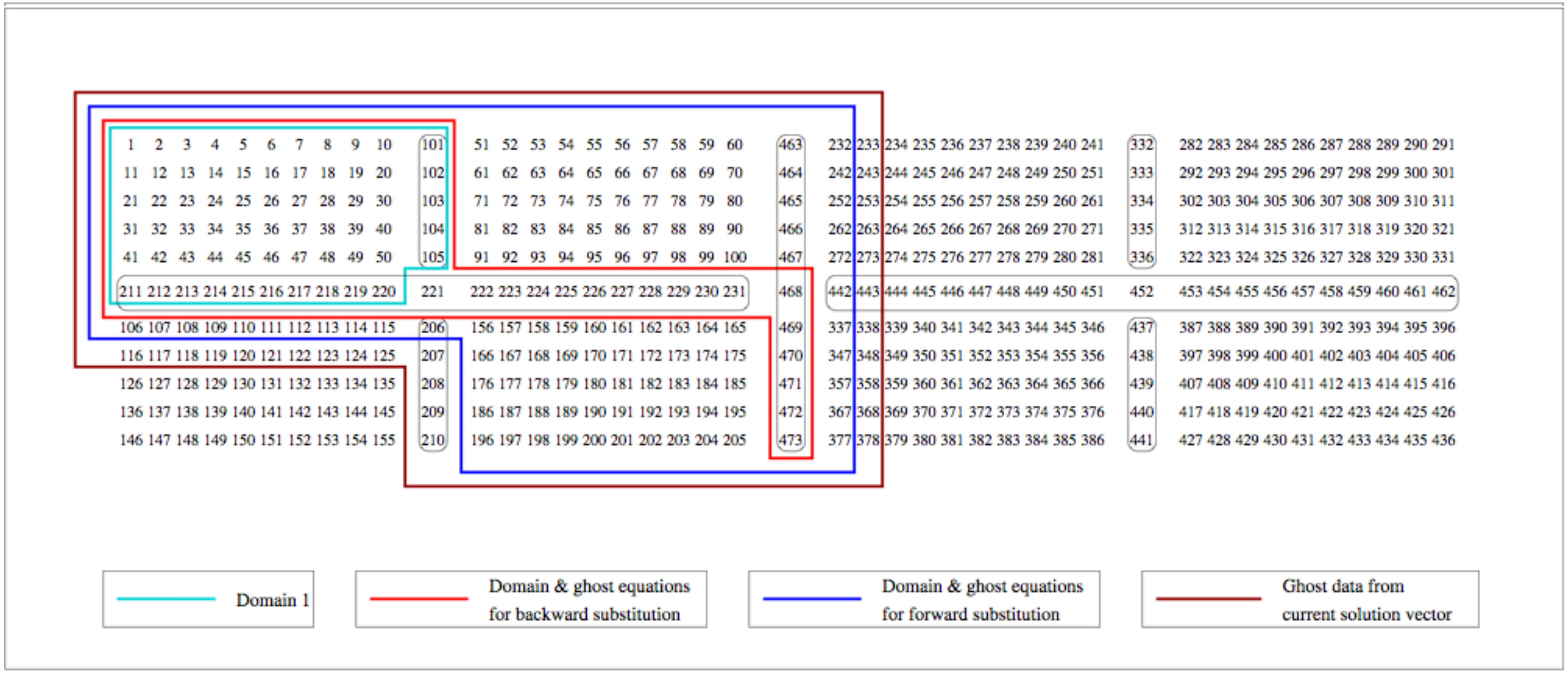
- Efficient preconditioners that reduce the number of iterations remain crucial

ILU0 with nested dissection and ghosting

Let α_0 be the set of equations to be solved by one processor
For $j = 1$ to s do
    Find β_j = *ReachableVertices* $(G(U), \alpha_{j-1})$
    Find γ_j = *ReachableVertices* $(G(L), \beta_j)$
    Find δ_j = *Adj* $(G(A), \gamma_j)$
    Set $\alpha_j = \delta_j$
end

Ghost data required:
$x(\delta)$, $A(\gamma,\delta)$,
$L(\gamma,\gamma)$, $U(\beta, \beta)$

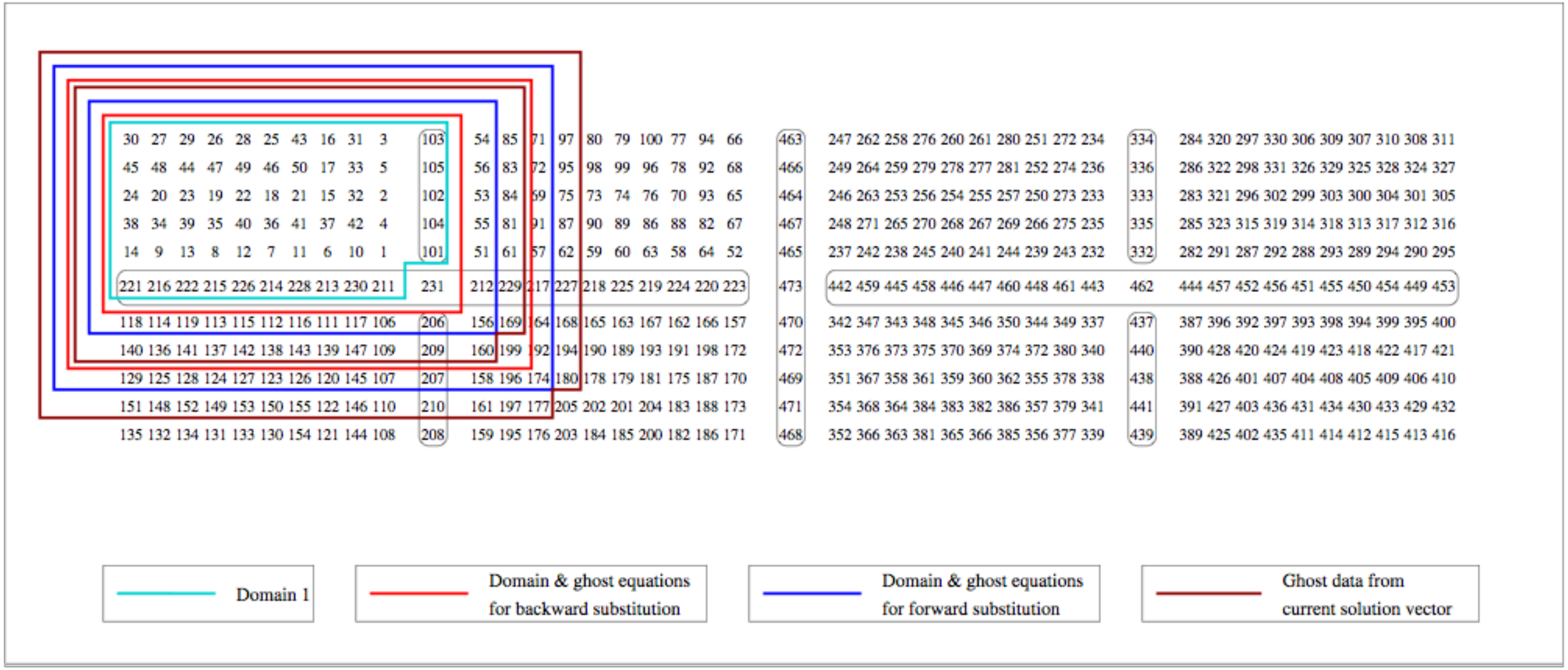
$\Rightarrow$ Half of the work performed on one processor



5 point stencil on a 2D grid

CA-ILU0 with alternating reordering and ghosting

- Reduce volume of ghost data by reordering the vertices:
  - First number the vertices at odd distance from the separators
  - Then number the vertices at even distance from the separators
- CA-ILU0 computes a standard ILU0 factorization



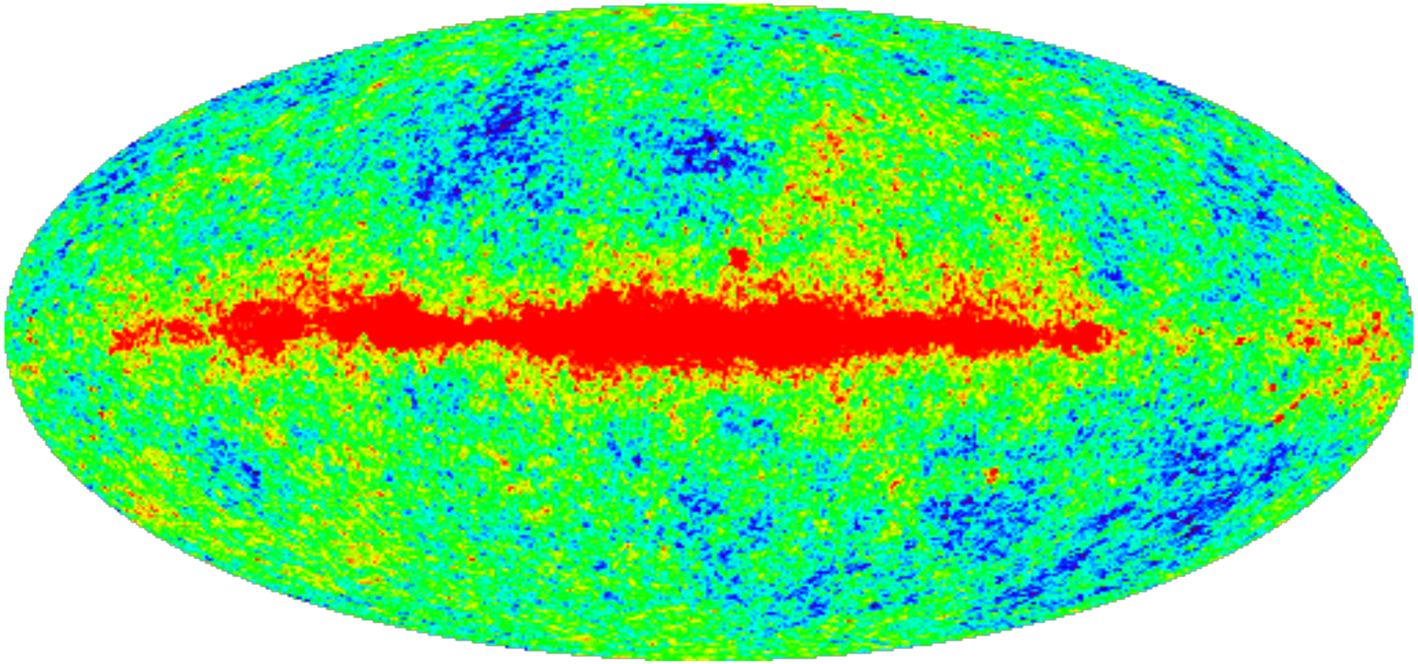
5 point stencil on a 2D grid

Plan

- Motivation
- Communication avoiding for numerical linear algebra
  - Novel algorithms that minimize communication
  - Often not in ScaLAPACK or LAPACK (YET !)
  - Iterative methods and preconditioning
- Application to CMB data analysis in astrophysics
- Conclusions

CMB data analysis

- Light left over after the ever mysterious «Big Bang»,
  - overall very isotropic and uniform,
  - but small - 1 part in 10^5 - anisotropies are hidden in there …
  - even smaller - 1 part in 10^6 or 10^7 - are the goal of current experiments.



- Always in need of more data
- Data sets are growing at Moore's rate

CMB data analysis in an (algebraic) nutshell

- CMB DA is a juxtaposition of the same algebraic operations
- Map-making problem
  - Find the best map x from observations d, scanning strategy A, and noise n_t

$$d = Ax + n_t$$

  - Assuming the noise properties are Gaussian and piece-wise stationary, the covariance matrix is $N = \langle n_t n_t^T \rangle$, and N^{-1} is a block diagonal symmetric Toeplitz matrix.
  - The solution of the generalized least squares problem is found by solving

$$A^T N^{-1} A x = A^T N^{-1} d$$

- Spherical harmonic transform (SHT)
  - Synthesize a sky image from its harmonic representation

- What is difficult about the CMB DA then ? Well, the data is BIG !
- Our solution to this challenge: MIDAPACK (ANR MIDAS interdisciplinary project)
  - Library implementing all the stages down the CMB pipeline
  - Results in collaboration with M. Szydlarski, R. Stompor (SC'12)

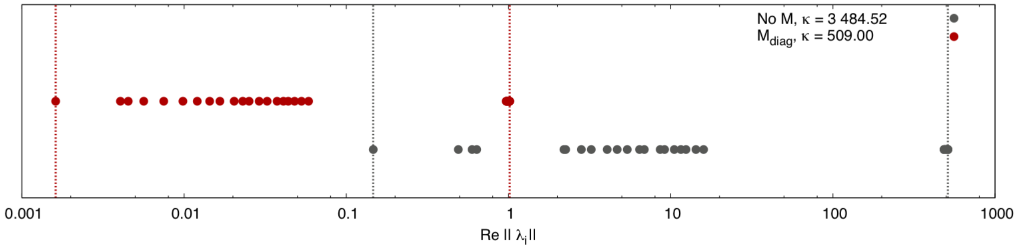
Challenge in the map-making problem

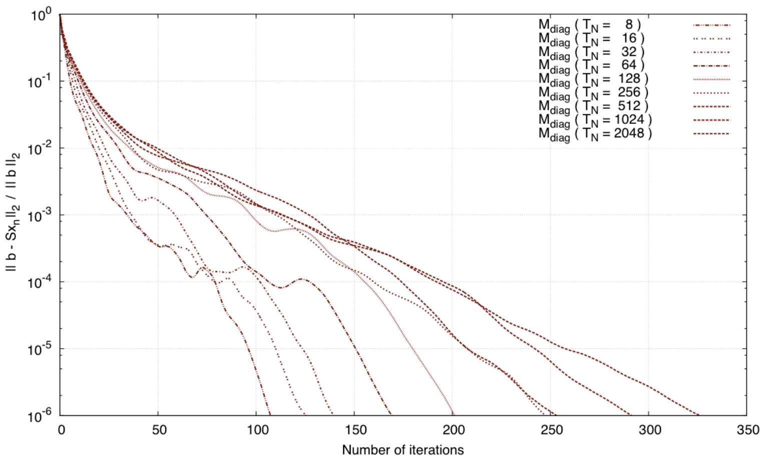
- Linear system to solve using PCG:

$$M_{diag} S x = M_{diag} b, \text{ where } \begin{cases} S := A^T N^{-1} A,\ b := A^T N^{-1} d \\ M_{diag} := \left(A^T diag(N^{-1}) A \right)^{-1} \end{cases}$$

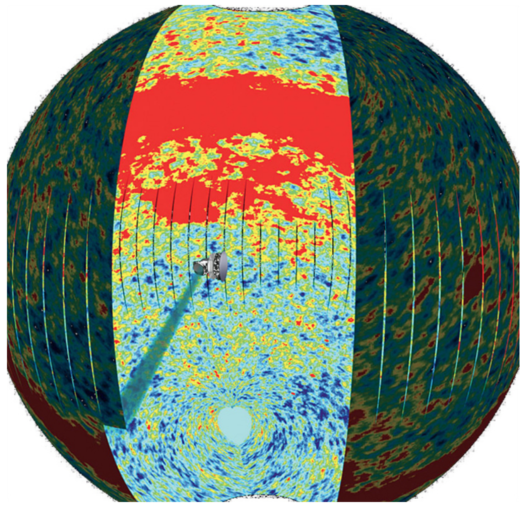
- Existing diagonal preconditioner does not scale numerically
- The convergence of iterative methods depends on the condition number of the input matrix - low eigenvalues hamper this convergence

Spectrum: 20 largest and 20 smallest approximated eigenvalues





Scanning strategy:
- 2048 densely crossing circles
- Each circle is scanned 32 times, leading to 10^6 samples
- Piece-wise stationary noise, one Toeplitz block for each circle



Two level preconditioner

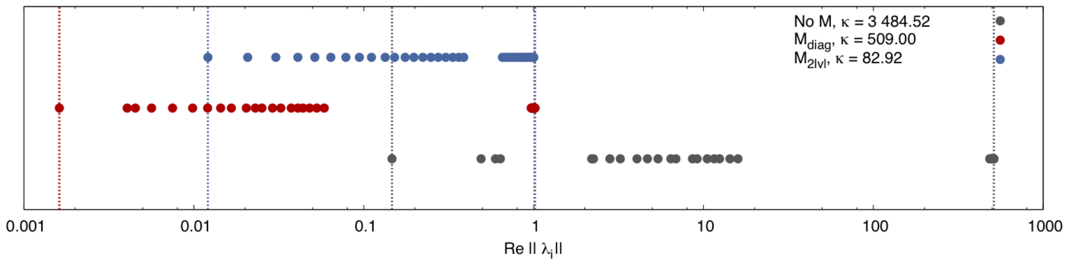
- Combine diagonal preconditioner with a subspace correction (Tang et al, 09)

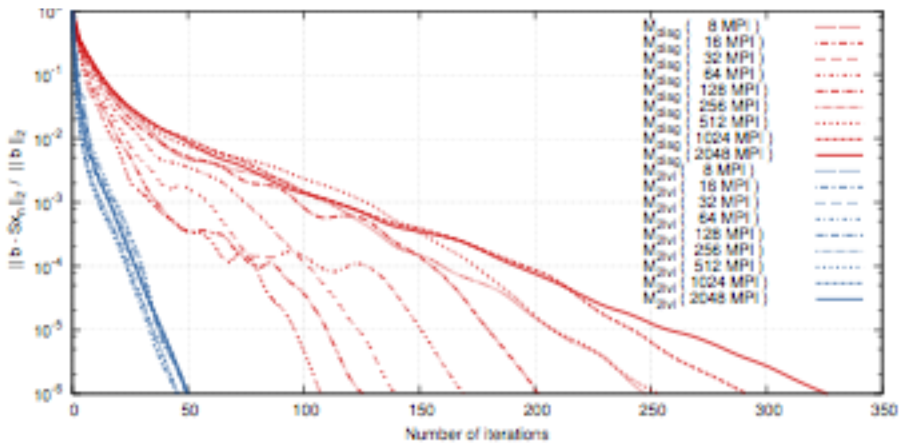
$$M_{2lvl} = M_{diag}\left(I - S(ZE^{-1}Z^T)\right) + \left(ZE^{-1}Z^T\right)$$

$$\text{where } M_{diag} = \left(A^T diag(N^{-1})A\right)^{-1} \text{ and } E = Z^T SZ$$

- The efficiency of the preconditioner depends on the choice of Z
  - Common approaches exist in deflation or coarse grid correction in DDM
  - Our choice is inspired by the physics of the CMB

Spectrum: 20 largest and 20 smallest approximated eigenvalues





Choice of coarse weighting subspace Z

- Number of columns of Z equals number of time-stationary intervals
- Each row of Z corresponds to a pixel of the sky, $Z(i,j) = s_i^j / s_i$, where
  - s_i^j is the number of observations of pixel i during j-th time interval
  - s_i is the total number of observations of pixel i.

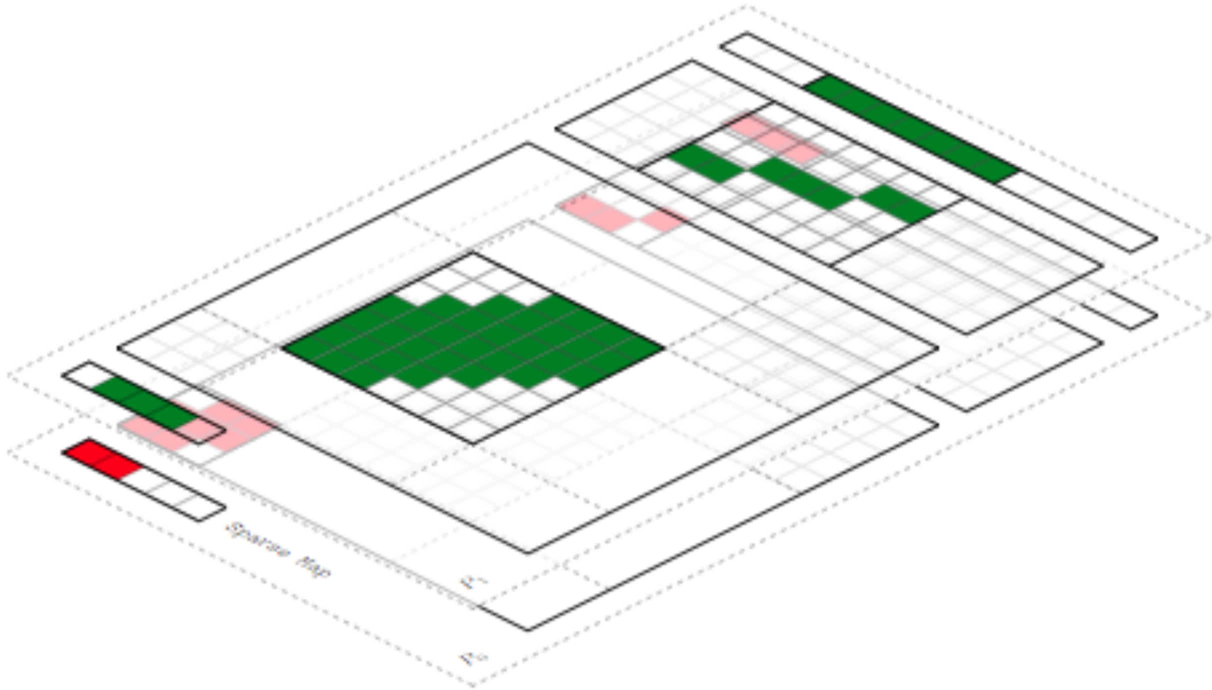
$$Z = \begin{pmatrix} \frac{s_0^0}{s_0} & \frac{s_0^1}{s_0} & \cdots & \frac{s_0^k}{s_0} \\ \frac{s_1^0}{s_1} & \frac{s_1^1}{s_1} & \cdots & \frac{s_1^k}{s_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{s_p^0}{s_p} & \frac{s_p^1}{s_p} & \cdots & \frac{s_p^k}{s_p} \end{pmatrix}$$

- Example:

$$\tilde{x} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix}, A^T = \left[\begin{array}{cc|ccc|cc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right], Z = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

Data distribution

- Block row distribution over processors of
  - Pointing matrix A, noise covariance matrix N^{-1},
  - Observations vector d and map of the sky x



Application of two-level preconditioner to a vector

$$\left[M_{diag}\left(I - SZE^{-1}Z^T\right) + \left(\mathrm{ZE}^{-1}Z^T\right)\right]v_{in} = v_{out}$$

The operations peformed are:

1. $v_{tmp1} := Z^T v_{in}$
   - Series of dot products followed by
   - MPI_AllReduce (…) <= the most expensive operation

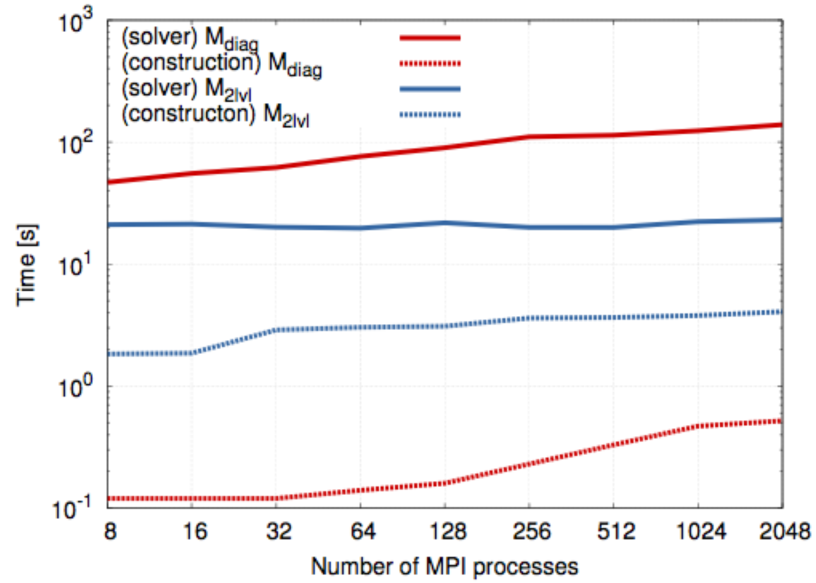
2. Solve $Ev_{tmp2} := v_{tmp1}$
   - Using direct solver as MKL, SuperLU

3. $v_{out} \mathrel{+}= Z\, v_{tmp2}$
   $v_{tmp3} := v_{in} - S\, Z\, v_{tmp2}$
   - Series of scalar vector producs

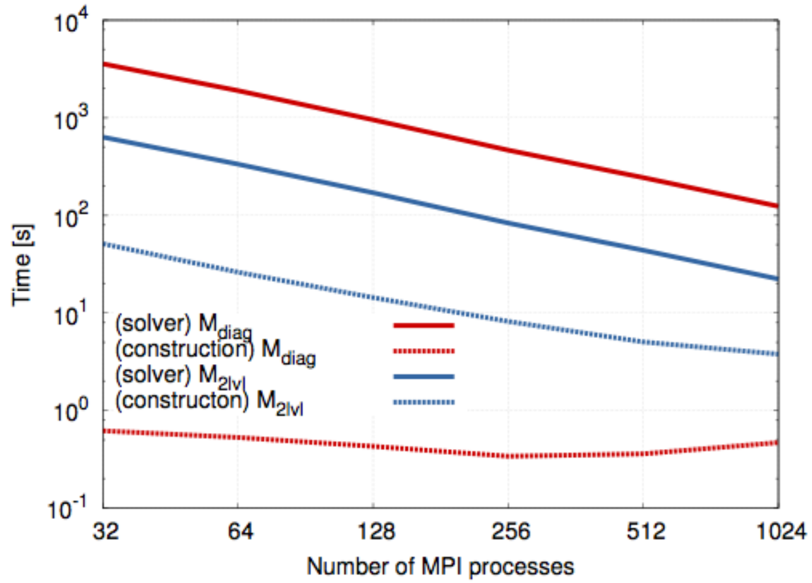
4. $v_{out} \mathrel{+}= M_{diag}\, v_{tmp3}$
   - entrywise product between two vectors

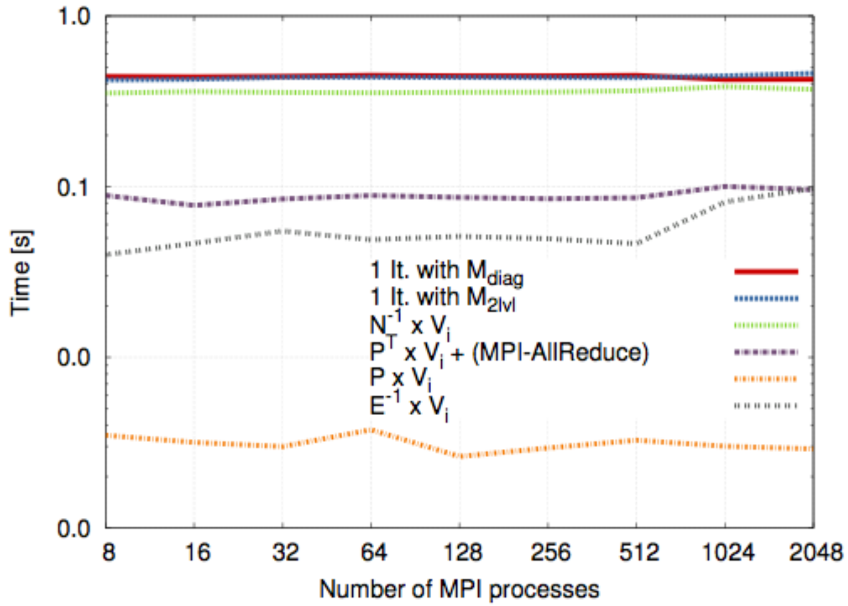
Runtime on Cray XE6, Hopper Nersc

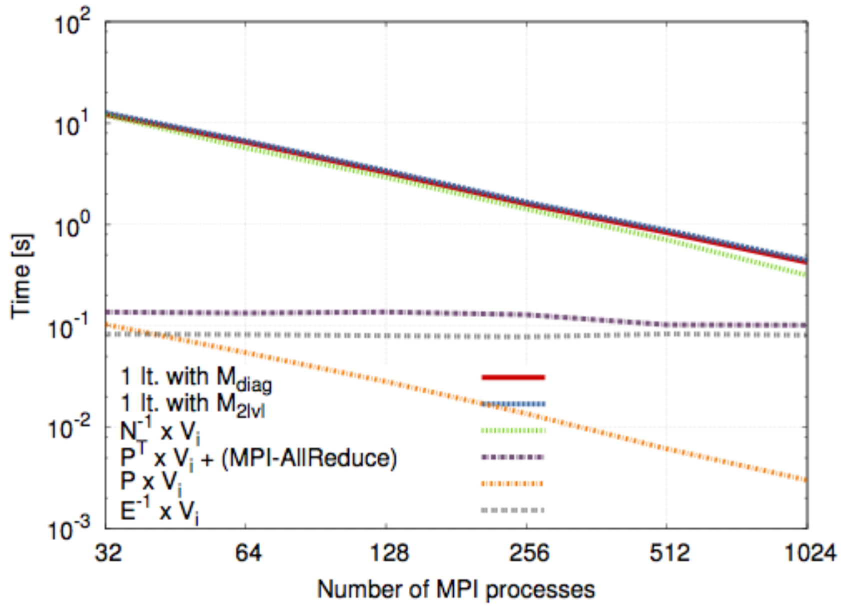
Weak scaling, 6 cores per MPI process



Strong scaling, 6 cores per MPI process

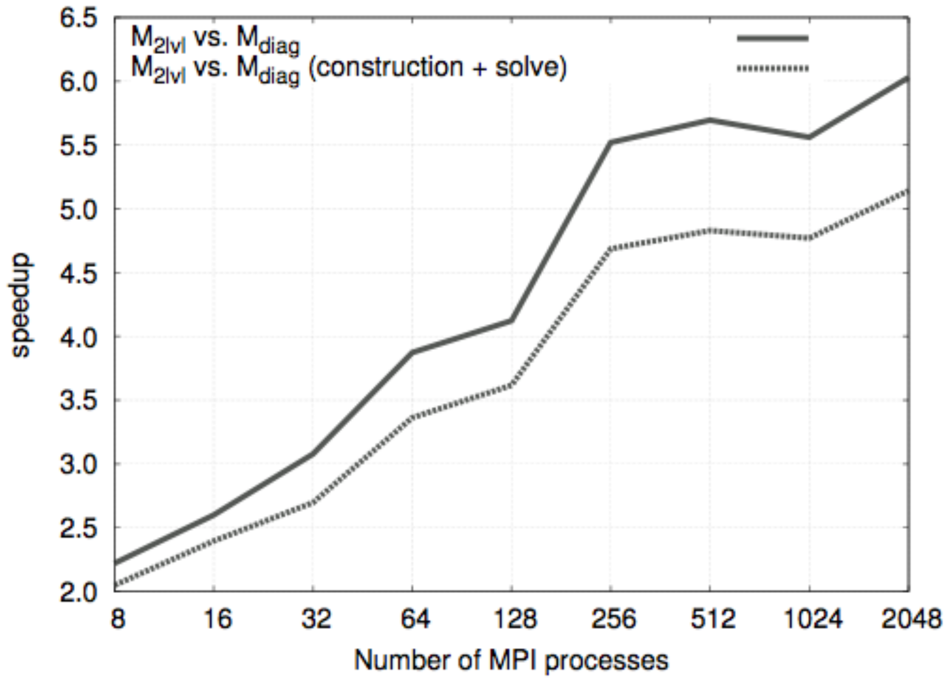




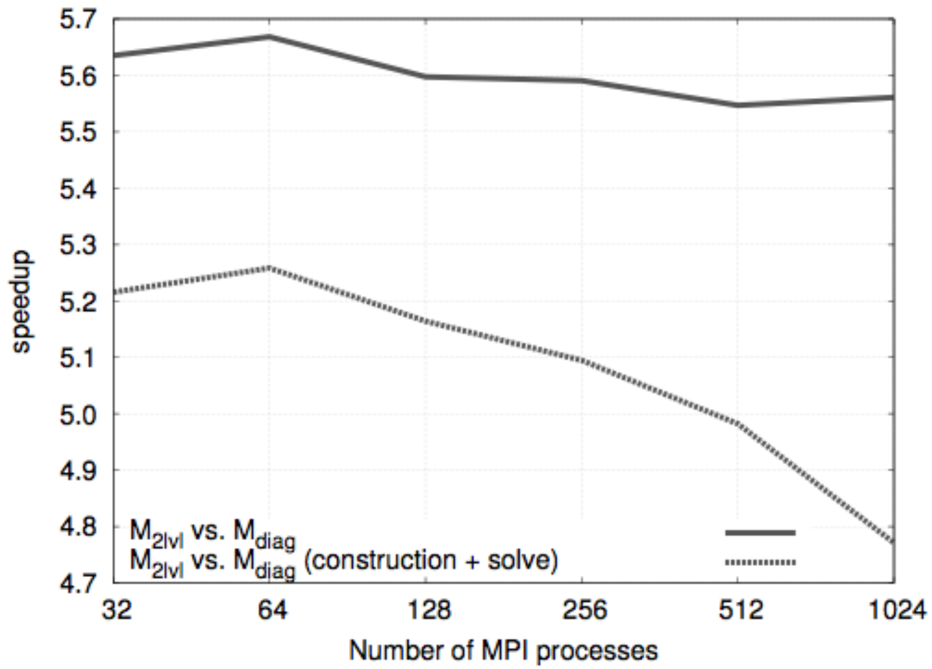


Improvement with respect to diagonal preconditioner

Weak scaling, 6 cores per MPI process



Strong scaling, 6 cores per MPI process



Conclusions

- Communication avoiding algorithms minimize communication
  - Attain theoretical lower bounds on communication
  - Are often faster than conventional algorithms in practice
- Remains a lot to do for sparse linear algebra
  - Communication bounds, communication optimal algorithms
  - Numerical stability of s-step methods
  - Preconditioners - limited by the memory size, not flops
- In CMB data analysis
  - Can we use randomized approaches ?

Collaborators, funding

Collaborators:

- INRIA: A. Branescu, S. Donfack, A. Khabou, M. Jacquelin, S. Moufawad, H. Xiang, M. Szydlarski, M. Shariffy
- J. Demmel, UC Berkeley, B. Gropp, UIUC, M. Gu, UC Berkeley, M. Hoemmen, UC Berkeley, J. Langou, CU Denver, V. Kale, UIUC, R. Stompor, Paris 7

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:

- P. Cargemel, L. Grigori, R. Stompor, Study of communication patterns of CMB data analysis, in preparation.
- 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.
- L. Grigori, R. Stompor, and M. Szydlarski, A two-level preconditioner for Cosmic Microwave Background map-making, SuperComputing 2012.
- 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.
- L. Grigori, S. Moufawad, *Communication avoiding incomplete LU preconditioner*, in preparation, 2012