Enhancing Scalability and Robustness of the Schur Complement Method Using Hierarchical Parallelism

Xiaoye Sherry Li
Lawrence Berkeley National Laboratory

Ichitaro Yamazaki
University of Tennessee, Knoxville

12th Copper Mountain Conference on Iterative Methods
March 25-30, 2012

Acknowledgements

Funded through DOE SciDAC projects:

<u>Solvers:</u>

- **TOPS (Towards Optimal Petascale Simulations)**
- **FASTMath (Frameworks, Algorithms, and Scalable Technologies for Mathematics)**

<u>Application partnership:</u>

- **CEMM (Center for Extended MHD Modeling, fusion energy)**
- **ComPASS (Community Petascale Project for Accelerator Science and Simulation)**

OUTLINE OF THE TALK

- **Hybrid solver based on Schur complement method**
  - Software: **PDSLin** (Parallel Domain decomposition Schur complement Linear solver)

- **Combinatorial problems in hybrid solver**
  - Sparse triangular solution with sparse right-hand sides
  - K-way graph partitioning with multiple constraints

… Focusing on scalability issues.

Schur complement method

- **a.k.a iterative substructuring method**
  or, non-overlapping domain decomposition

- **Divide-and-conquer paradigm . . .**
  - Divide entire problem (domain, graph) into subproblems (subdomains, subgraphs)
  - Solve the subproblems
  - Solve the interface problem (Schur complement)

- **Variety of ways to solve subdomain problems and Schur complement … lead to a powerful poly-algorithm or hybrid solver framework**

Algebraic view

1. **Reorder into 2x2 block system, A_{11} is block diagonal**

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

2. **Schur complement**

$$S = A_{22} - A_{21} A_{11}^{-1} A_{12} = A_{22} - (U_{11}^{-T} A_{21}^{T})^{T} (L_{11}^{-1} A_{12}) = A_{22} - W \cdot G$$

$$\text{where} \quad A_{11} = L_{11} U_{11}$$

S corresponds to interface (separator) variables, no need to form explicitly

3. **Compute the solution**

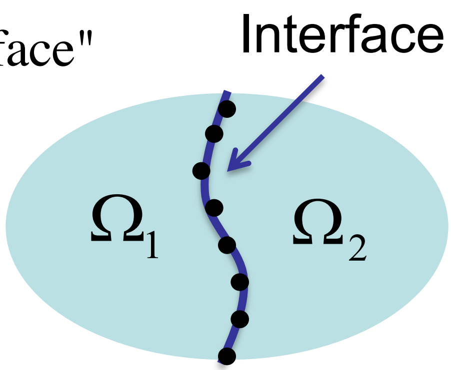
$$(1) \quad x_2 = S^{-1}(b_2 - A_{21} A_{11}^{-1} b_1) \quad \leftarrow \quad \text{iterative solver}$$

$$(2) \quad x_1 = A_{11}^{-1}(b_1 - A_{12} x_2) \quad \leftarrow \quad \text{direct solver}$$

Structural analysis view

- **Case of two subdomains**

Substructure contribution: $A^{(k)} = \begin{pmatrix} A_{ii}^{(k)} & A_{iI}^{(k)} \\ A_{Ii}^{(k)} & A_{II}^{(k)} \end{pmatrix}$ $\quad i = \text{"interior"}$, $I = \text{"Interface"}$



1. Assembled block matrix $A = \begin{pmatrix} A_{ii}^{(1)} & & A_{iI}^{(1)} \\ & A_{ii}^{(2)} & A_{iI}^{(2)} \\ A_{Ii}^{(1)} & A_{Ii}^{(2)} & A_{II}^{(1)} + A_{II}^{(2)} \end{pmatrix}$

2. Perform direct elimination of $A^{(1)}$ and $A^{(2)}$ independently,

   Local Schur complements: $S^{(k)} = A_{II}^{(k)} - A_{Ii}^{(k)} (A_{ii}^{(k)})^{-1} A_{iI}^{(k)}$

   Assembled Global Schur complement $S = S^{(1)} + S^{(2)}$

Solving the Schur complement system

- **SPD, conditioning property** [Smith/Bjorstad/Gropp'96]

  For an SPD matrix, condition number of a Schur complement is no larger than that of the original matrix.

  - S is SPD, much reduced in size, better conditioned, but denser, solvable with preconditioned iterative solver

- **Two approaches to preconditioning S**

  1. Global S (e.g., PDSLin [Yamazaki/L.'10], HIPS [Henon/Saad'08])
     - **general algebraic preconditioner, e.g. ILU(S)**
  2. Local S (e.g. MaPHys [Giraud/Haidary/Pralet'09])
     - **restricted preconditioner; more parallel**
     - **e.g., additive Schwarz preconditioner** $S = S^{(1)} \oplus S^{(2)} \oplus S^{(3)} \ldots$

$$M = {S^{(1)}}^{-1} \oplus {S^{(2)}}^{-1} \oplus {S^{(3)}}^{-1} \ldots$$

Related work

PDSLin (LBNL)	MaPHyS (INRIA/CERFACS)	HIPS (INRIA)
Multiple procs/subdom, Smaller Schur compl.	Multiple procs/subdom, Smaller Schur compl.	Multiple subdoms/proc, Larger Schur compl., Good load balance.
Threshold based “ILU”, Global approx. Schur Robust precond.	Additive Schwartz, Local Schur Scalable precond.	Block level-based ILU, Global approx. Schur Scalable precond.

- **PDSLin**
  - Uses two levels of parallelization and load-balancing techniques for tackling large-scale systems
  - Provides a robust preconditioner for solving highly-indefinite or ill-conditioned systems
- **On-going work:** comparison of PDSLin and MaPHyS [Yamazaki et al.]

Parallelization with serial subdomain

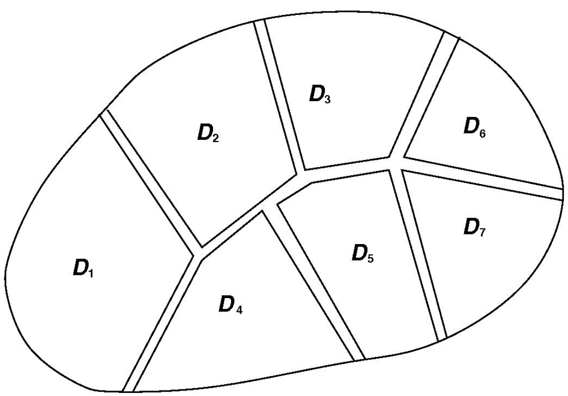
- **No. of subdomains increases with increasing core count.**
  ➔ Schur complement size and iteration count increase

- **HIPS (serial subdomain) vs. PDSLin (parallel subdomain)**
  - M3D-C[1], Extended MHD to model fusion reactor tokamak, 2D slice of 3D torus
  - Dimension 801k, 70 nonzeros per row, real unsymmetric

P	N_S	HIPS 1.0 sec (iter)	PDSLin sec (iter)
8	13k	284.6 (26)	79.9 (15)
32	29k	55.4 (64)	25.3 (16)
128	62k	--	17.1 (16)
512	124k	--	21.9 (17)

Parallelism – extraction of multiple subdomains

- **Partition adjacency graph of $|A|+|A^T|$**
  Multiple goals: reduce size of separator, balance size of subdomains
  - nested dissection (e.g., PT-Scotch, ParMetis)
  - k-way partition (preferred)

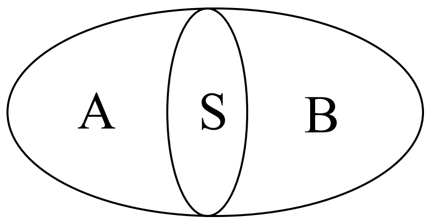


$$\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) = \left(\begin{array}{cccc|c} D_1 & & & & E_1 \\ & D_2 & & & E_2 \\ & & \ddots & & \vdots \\ & & & D_k & E_k \\ \hline F_1 & F_2 & \ldots & F_k & A_{22} \end{array}\right)$$

- **Memory requirement: fill is restricted within**
  - “small” diagonal blocks of A_{11}, and
  - ILU(S), maintain sparsity via numerical dropping

Graph partitioning

- Generalized nested dissection [Lipton/Rose/Tarjan '79]
  - **Top-down, divide-and-conquer, best for largest problems**
  - **Parallel codes : ParMetis, PT-Scotch**
  - **First level**



$$\longrightarrow \begin{bmatrix} A & 0 & x \\ 0 & B & x \\ x & x & S \end{bmatrix}$$

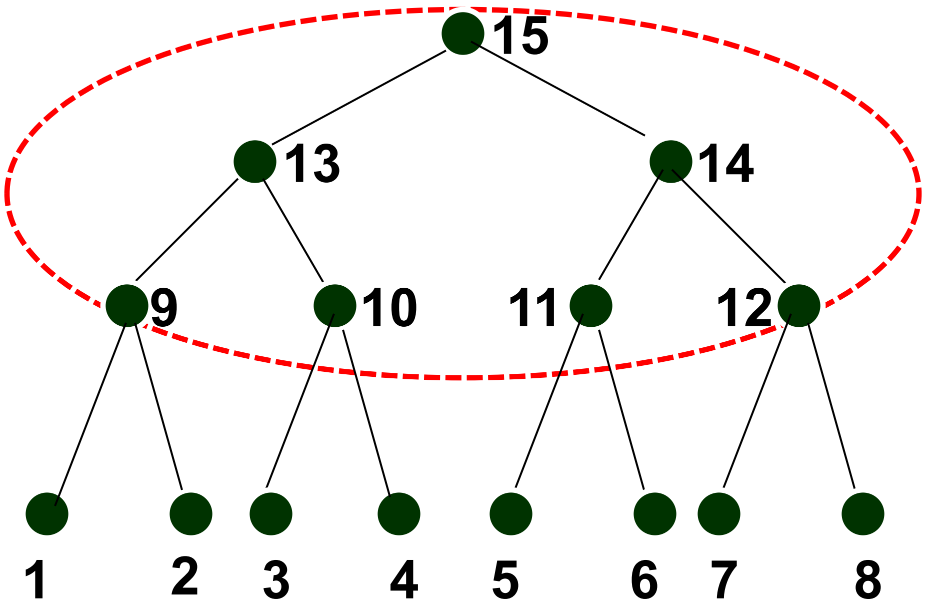
  - **Recurse on A and B**

- Goal: find the smallest separator S at each level
  - **Multilevel schemes:**
    - **ParMetis [Karypis et al.], PT-Scotch [Pellegrini et al.], PaToh [Catalyurek et al.]**
  - **Spectral bisection [Simon et al. `90-`95]**
  - **Geometric and spectral bisection [Chan/Gilbert/Teng `94]**

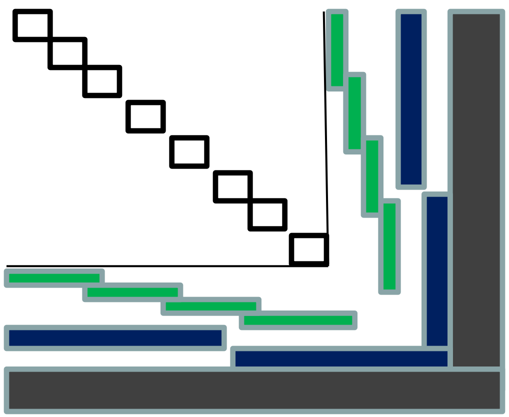
Ordering

- **Permute all the separators to the end**

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \left(\begin{array}{cccc|c} D_1 & & & & E_1 \\ & D_2 & & & E_2 \\ & & \ddots & & \vdots \\ & & & D_k & E_k \\ \hline F_1 & F_2 & \dots & F_k & A_{22} \end{array}\right)$$

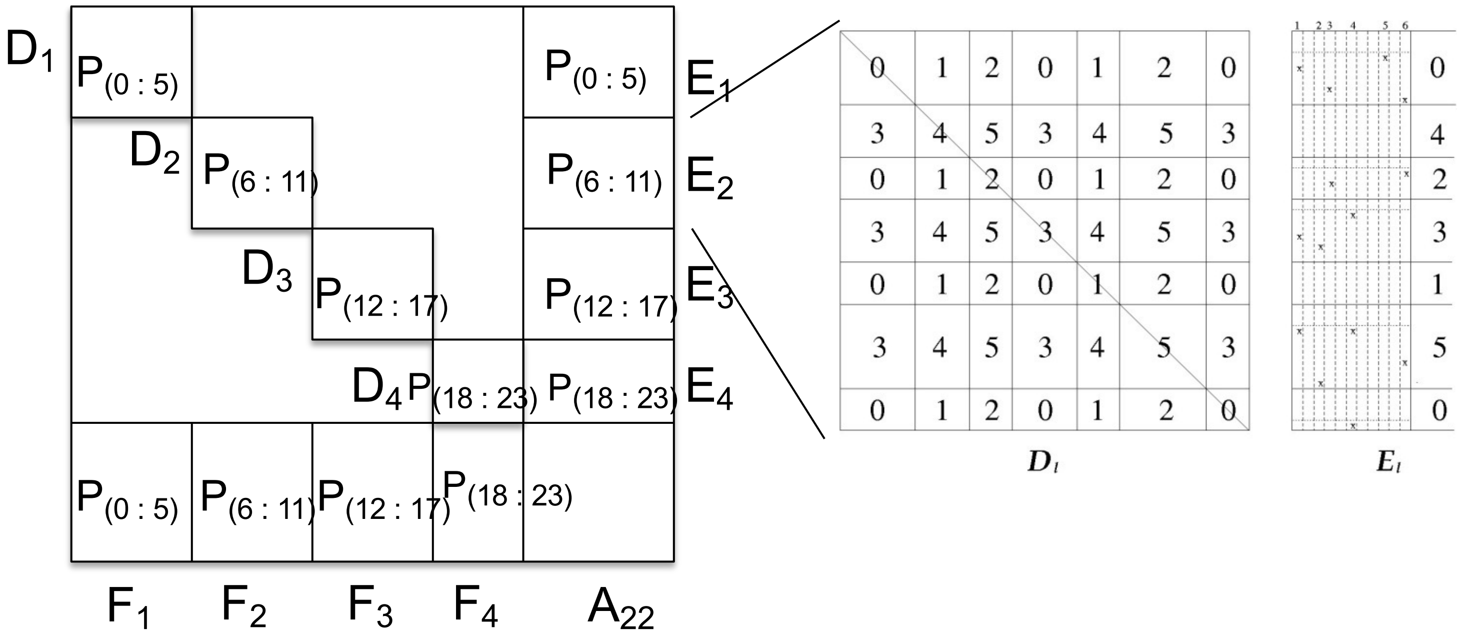


Separator tree



Hierarchical parallelism

- **Multiple processors per subdomain**
  - one subdomain with 2x3 procs (e.g. SuperLU_DIST, MUMPS)



- **Advantages:**
  - Constant #subdomains, Schur size, and convergence rate, regardless of core count.
  - Need only modest level of parallelism from direct solver.

PDSLin software

- **Parallel Domain decomposition Schur complement based Linear solver**
  - C and MPI, with Fortran interface.
  - Unsymmetric and symmetric, real and complex, multiple RHSs.
- **Requires the following external packages:**
  - parallel graph partitioning:
    - **PT-Scotch (http://labri.fr/perso/pelegrin/scotch), or**
    - **ParMetis (http://glaros.dtc.umn.edu/gkhome/views/metis)**
  - subdomain solver options:
    - **SuperLU (http://crd.lbl.gov/˜xiaoye/SuperLU)**
    - **SuperLU_DIST or SuperLU_MT: two levels of parallelization**
    - **MUMPS**
    - **PDSLin**
    - **Inexact subdomain solution: ILU**
  - Schur complement solver options:
    - **PETSc (http://mcs.anl.gov/petsc/petsc-as)**
    - **SuperLU_DIST**

PDSLin encompass Hybrid, Iterative, Direct

Hybrid

Default
Subdomain: LU
Schur: Krylov

Iterative

Options
(1) num_doms = 0
Schur = A: Krylov
(2) FGMRES Inner-Outer:
Subdomain: ILU
Schur: Krylov

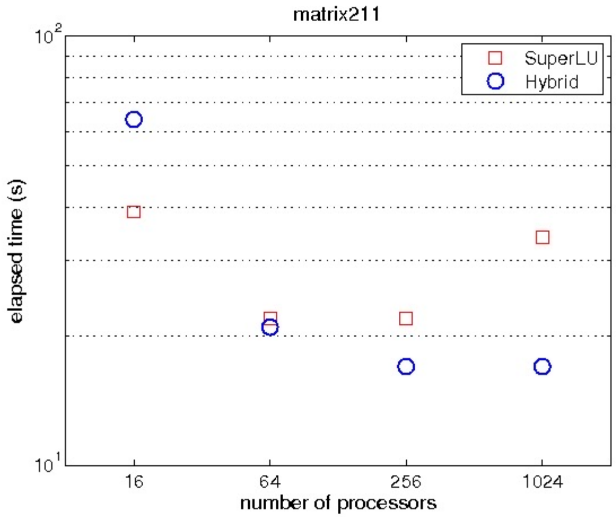
Direct

Options
(1) Subdomain: LU
Schur: LU
drop_tol = 0.0
(2) num_doms = 1
Domain: LU

$$\left(\begin{array}{cccc|c} D_1 & & & & E_1 \\ & D_2 & & & E_2 \\ & & \ddots & & \vdots \\ & & & D_k & E_k \\ \hline F_1 & F_2 & \dots & F_k & A_{22} \end{array}\right)$$

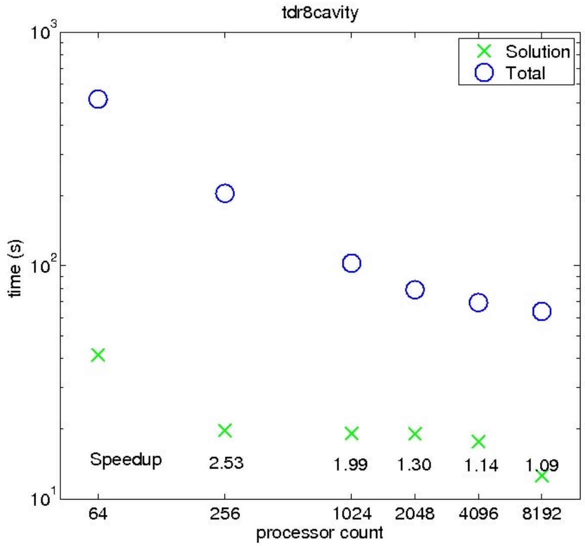
PDSLin vs. SuperLU_DIST

- **Cray XT4 at NERSC**
- **Matrix211 : extended MHD to model burning plasma**
  - dimension = 801K, nonzeros = 56M, real, unsymmetric
  - PT-Scotch extracts 8 subdomains of size ≈ 99K, S of size ≈ 13K
  - SuperLU_DIST to factorize each subdomain, and compute preconditioner LU($\widetilde{S}$)
  - BiCGStab of PETSc to solve Schur system on 64 processors with residual < 10^{-12} , converged in 10 iterations
- **Needs only 1/3 memory of direct solver**



PDSLin for RF cavity (strong scaling)

- **Cray XT4 at NERSC; used 8192 cores**
- **Tdr8cavity : Maxwell equations to model cavity of International Linear Collider**
  - dimension = 17.8M, nonzeros = 727M
  - PT-Scotch extracts 64 subdomains of size ≈ 277K, S of size ≈ 57K
  - BiCGStab of PETSc to solve Schur system on 64 processors with residual < 10^{-12}, converged in 9 ~ 10 iterations
- **Direct solver failed !**



PDSLin for largest system

- Matrix properties:
  - 3D cavity design in Omega3P, 3rd order basis function for each matrix element
  - dimension = 52.7 M, nonzeros = 4.3 B (~80 nonzeros per row), real, symmetric, highly indefinite
- Experimental setup:
  - PT-Scotch extracts 128 subdomains of size ~410k
  - SuperLU DIST factors each subdomain, computes preconditioner LU($\widetilde{S}$) of size 247k (32 cores)
  - BiCGStab of PETSc to solve Sy = d
- Performance:
  - Fill-ratio (nnz(Precond.)/nnz(A)): ~ 250
  - Using 2048 cores:
    - preconditioner construction: 493.1 sec.
    - solution: 108.1 second (32 iterations)

Combinatorial problems

[Yamazaki, L., Rouet, Ucar]

Computing approximate Schur preconditioner

Combinatorial problems . . .

- **Sparse triangular solution with many sparse RHS**

$$S = A_{22} - \sum_{l} (U_l^{-T} F_l^T)^T \, (L_l^{-1} E_l), \text{ where } D_l = L_l U_l$$

- **Sparse matrix–matrix multiplication**

$$\widetilde{G} \leftarrow \text{sparsify}(G, \sigma_1); \quad \widetilde{W} \leftarrow \text{sparsify}(W, \ \sigma_1)$$

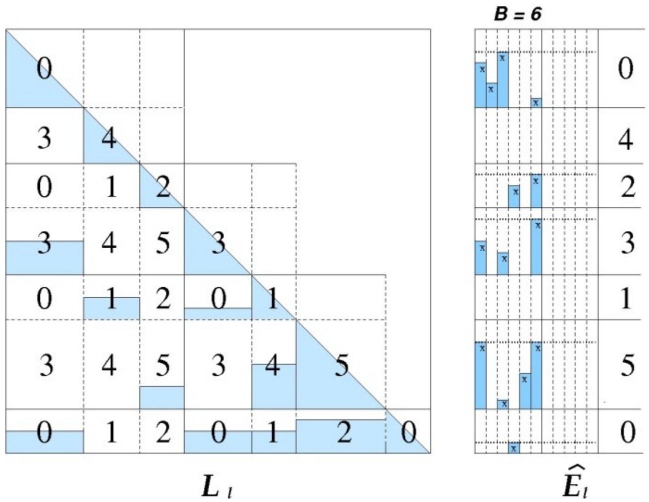
$$T^{(p)} \leftarrow \widetilde{W}^{(p)} \times \widetilde{G}^{(p)}$$

$$\hat{S}^{(p)} \leftarrow A_{22}{}^{(p)} - \sum\nolimits_q T^{(q)}(p); \quad \widetilde{S} \leftarrow \text{sparsify}(\hat{S}, \ \sigma_2)$$

- **K-way graph partitioning with multiple objectives**
  - Small separator
  - Similar subdomains
  - Similar connectivity

1. Sparse triangular solution with sparse RHS

- **RHS vectors E_ℓ and F_ℓ are sparse (e.g., about 20 nnz per column); There are many RHS vectors (e.g., $O(10^4)$ columns)**

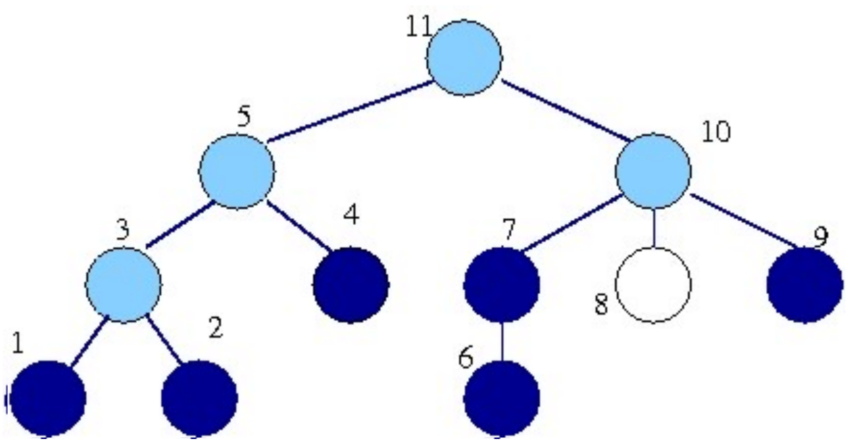


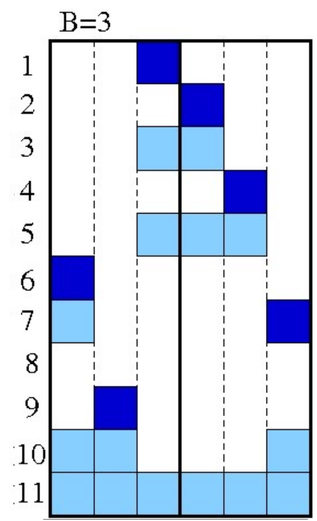
- **Blocking the RHS vectors**
  - Reduce number of calls to the symbolic routine and number of messages, and improve read reuse of the LU factors
  - ➢ Achieved over 5x speedup
  - ✖ zeros must be padded to fill the block

Sparse triangular solution with sparse RHSs

- **Combinatorial question: Reorder columns of E_ℓ to maximize structural similarity among the adjacent columns.**
- **Where are the fill-ins?**

  <u>Path Theorem</u> [Gilbert'94] **Given the elimination tree of D_l, fill will be generated in G_l at the positions associated with the nodes on the path from nodes of the nonzeros in E_l to the root**

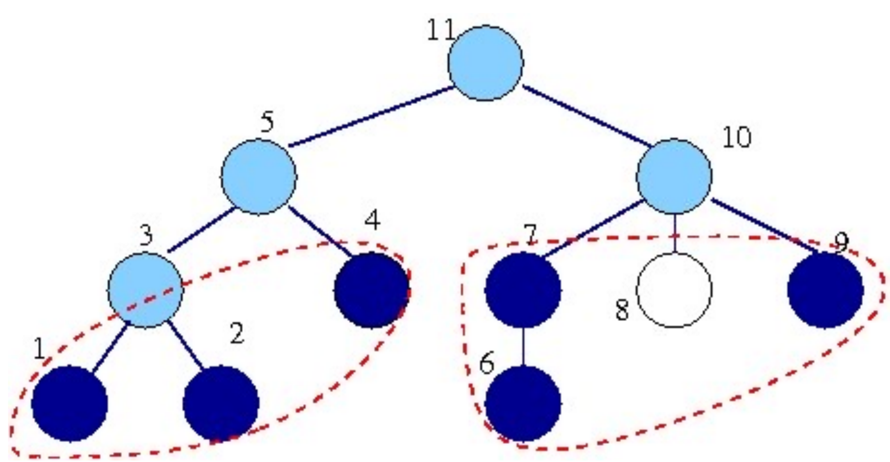


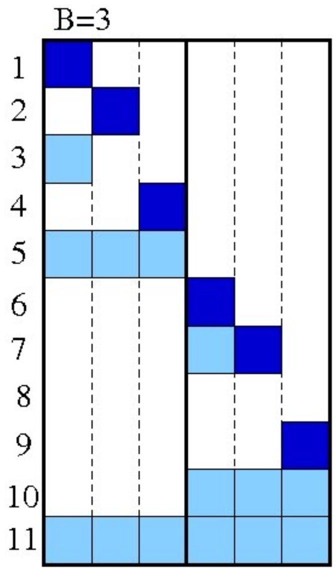


24 padded zeros

Sparse triangular solution … postordering

- **Postorder-conforming ordering of the RHS vectors**
  - Postorder the elimination tree
  - Permute the columns of E_I such that the row indices of the first nonzeros are in ascending order
- **Increased overlap of the paths to the root, fewer padded zeros**
- **20 – 40% reduction in solution time . . . improved over ND.**





13 padded zeros

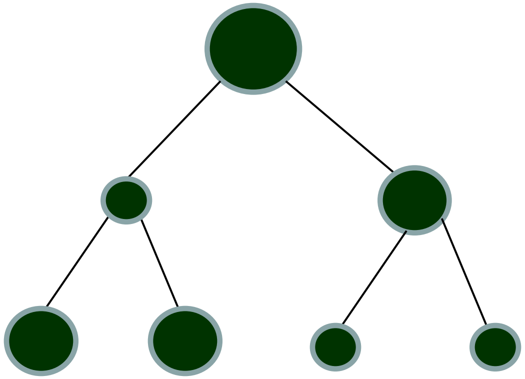
Sparse triangular solution … further optimization

- **Define a cost function that measures the dissimilarity of the sparsity pattern within a partition.**
- **Reordering based on a hypergraph partitioning model which minimizes the cost function above.**
- **Led to additional 10% reduction in time.**

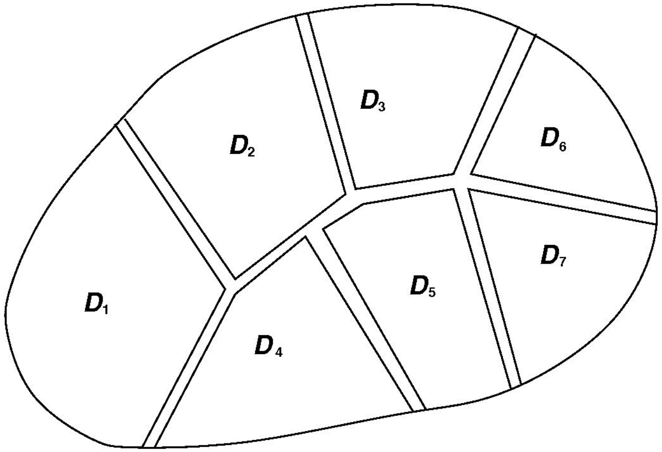
2. K-way subdomain extraction

- **Problem with ND:**

  Imbalance in separator size at different branches → Imbalance in subdomain size

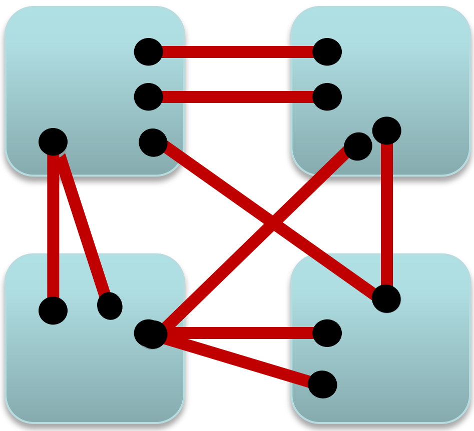


- **Alternative: directly partition into K parts, meet multiple constraints:**
  1. **Compute k-way partitioning → balanced subdomains**
  2. **Extract separator → balanced connectivity**



Experimenting several heuristics

- **Direct partition**
  1. Scotch: k-way edge partition
     - **Parts have similar size; minimize number of edge cuts**
  2. Form edge-induced subgraph
     - **Two end-points form wide vertex separator**
     - **Find smaller vertex separator via minimal vertex cover**



- **Recursive hypergraph partitioning**

- **Results**

+ improved balance of subdomains and interface
– larger Schur complement & interfaces
→ total time improved a little

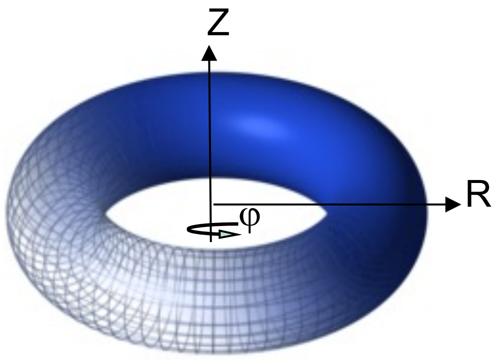
Final remarks

- **Direct solvers can scale to 1000s cores**
- **Domain-decomposition type of hybrid solvers can scale to 10,000s cores**
  - Can maintain robustness too
- **Expect to scale more with low-rank structured factorization preconditioner**

Extra Slides

Application 1: Burning plasma for fusion energy

- DOE SciDAC project: Center for Extended Magnetohydrodynamic Modeling (CEMM),   PI: S. Jardin, PPPL
- **Develop simulation codes to predict microscopic MHD instabilities of burning magnetized plasma in a confinement device (e.g., tokamak used in ITER experiments).**
  - Efficiency of the fusion configuration increases with the ratio of thermal and magnetic pressures, but the MHD instabilities are more likely with higher ratio.
- **Code suite includes M3D-C^1, NIMROD**



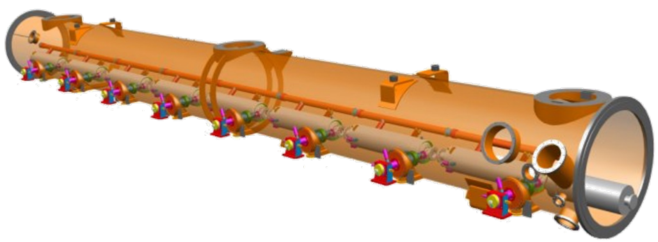
(S. Jardin)

- At each φ = constant plane, scalar 2D data is represented using 18 degree of freedom quintic triangular finite elements Q_{18}
- Coupling along toroidal direction

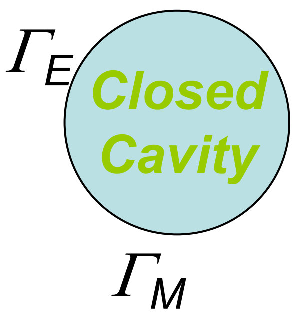
Application 2: Accelerator cavity design

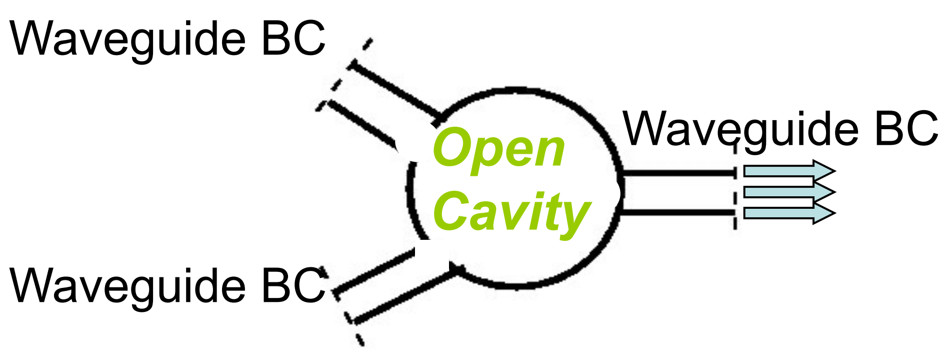
- DOE SciDAC: Community Petascale Project for Accelerator Science and Simulation (ComPASS), PI: P. Spentzouris, Fermilab
- **Development of a comprehensive computational infrastructure for accelerator modeling and optimization**
- **RF cavity: Maxwell equations in electromagnetic field**
- **FEM in frequency domain leads to large sparse eigenvalue problem; needs to solve shifted linear systems**

(L.-Q. Lee)



RF unit in ILC





linear eigenvalue problem

$$(K_0 - \sigma^2 M_0)\, x = M_0\, b$$

nonlinear complex eigenvalue problem

$$(K_0 + i\,\sigma W - \sigma^2 M_0)\, x = b$$