A High-Performance Parallel Hybrid Method for Large Sparse Linear Systems

Azzam Haidar

CERFACS, Toulouse

joint work with

Luc Giraud (N7-IRIT, France) and Layne Watson (Virginia Polytechnic Institute, USA)

Sparse Days Meeting 2007

October 10th-12th, 2007, CERFACS, Toulouse, France

Outline

Outline

Motivations

Solution of very Large/Huge *ill*-conditioned linear systems

- Such problems can require thousands of CPU-hours and many Gigabytes of memory
- Direct solvers:
  - Robust and usually do not fail
  - Memory and computational costs grow nonlinearly
- Iterative solvers:
  - Reduce memory requirements
  - They may fail to converge
  - Typically implemented with preconditioning to accelerate convergence

In an effort to reduce these requirements, a parallel mechanism for combining solvers is needed

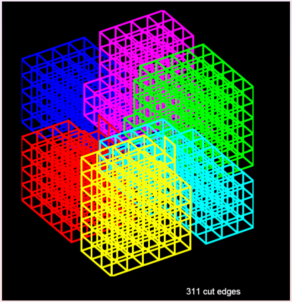
Goal

Develop a robust scalable parallel hybrid direct/iterative linear solvers

- Exploit the efficiency and robustness of the sparse direct solvers
- Take advantage of the natural scalable parallel implementation of iterative solvers
- Develop robust parallel preconditioners for iterative solvers

Non-overlapping domain decomposition

- Natural approach for PDE's
- Extend to general sparse matrices
- Partition the problem into subdomains, subgraphs
- Use a direct solver on the subdomains
- Robust preconditioned iterative solver on interface



Outline

Background

Algebraic splitting and block Gaussian elimination: N sub-domains case

$$\begin{pmatrix} A_{I_1 I_1} & \dots & 0 & A_{I_1 \Gamma_1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & A_{I_N I_N} & A_{I_N \Gamma_N} \\ A_{\Gamma_1 I_1} & \dots & A_{\Gamma_N I_N} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} u_{I_1} \\ \vdots \\ u_{I_N} \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} f_{I_1} \\ \vdots \\ f_{I_N} \\ f_\Gamma \end{pmatrix}$$

$$S u_\Gamma = \left(\sum_{i=1}^{N} R_{\Gamma_i}^T S^{(i)} R_{\Gamma_i} \right) u_\Gamma = f_\Gamma - \sum_{i=1}^{N} R_{\Gamma_i}^T A_{\Gamma_i I_i} A_{I_i I_i}^{-1} f_{I_i}$$

where $S^{(i)} = A_{\Gamma_i \Gamma_i}^{(i)} - A_{\Gamma_i I_i} A_{I_i I_i}^{-1} A_{I_i \Gamma_i}$

Spectral properties for elliptic PDE's

$$\kappa(A) = \mathcal{O}(h^{-2}) \qquad \kappa(S) = \mathcal{O}(h^{-1})$$

$$||e^{(k)}||_A \le 2 \cdot \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k ||e^{(0)}||_A$$

Outline

Additive Schwarz preconditioner [Carvalho, Giraud, Meurant, 01]

Preconditionner properties

- $$M_{AS} = \sum_{i=1}^{\#domains} R_i^T (\bar{S}^{(i)})^{-1} R_i$$

$$\bar{S}^{(i)} = \begin{pmatrix} S_{mm} & S_{mg} & S_{mk} & S_{m\ell} \\ S_{gm} & S_{gg} & S_{gk} & S_{g\ell} \\ S_{km} & S_{kg} & S_{kk} & S_{k\ell} \\ S_{\ell m} & S_{\ell g} & S_{\ell k} & S_{\ell\ell} \end{pmatrix} \qquad S^{(i)} = \begin{pmatrix} S_{mm}^{(i)} & S_{mg} & S_{mk} & S_{m\ell} \\ S_{gm} & S_{gg}^{(i)} & S_{gk} & S_{g\ell} \\ S_{km} & S_{kg} & S_{kk}^{(i)} & S_{k\ell} \\ S_{\ell m} & S_{\ell g} & S_{\ell k} & S_{\ell\ell}^{(i)} \end{pmatrix}$$

Assembled local Schur complement

local Schur complement

$$S_{mm} = \sum_{j \in adj(m)} S_{mm}^{(j)}$$

Parallel implementation for solving $Au = f$

- Each *subdomain* $A^{(i)}$ is handled by one *processor*

$$A^{(i)} \equiv \begin{pmatrix} A_{I_iI_i} & A_{I_i\Gamma_i} \\ A_{I_i\Gamma_i} & A_{\Gamma_i\Gamma_i} \end{pmatrix}$$

- Concurrent partial factorizations are performed on each processor to form the so called "local Schur complement"

$$S^{(i)} = A^{(i)}_{\Gamma_i\Gamma_i} - A_{\Gamma_iI_i}A^{-1}_{I_iI_i}A_{I_i\Gamma_i}$$

- The reduced system $Sx = b$ is solved using a distributed Krylov solver
  - One matrix vector product per iteration each processor compute $S^{(i)}(x^{(i)})^k = (y^{(i)})^k$
  - One local preconditioner apply $(\bar{S}^{(i)})^{-1}(z^{(i)})^k = (r^{(i)})^k$
  - Local neighbor-neighbor communication per iteration
  - Dot products per iteration (reduction)
- Compute simultaneously the solution for the interior unknowns $A_{I_iI_i}u_{I_i} = f_{I_i} - A_{I_i\Gamma_i}u_{\Gamma_i}$

What tricks exist to construct cheaper preconditioners

Mixed arithmetic strategy

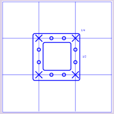
- Idea: Exploit 32 bit operation whenever possible and ressort to double at critical stages
- Compute and store the preconditioner in single precision arithmetic

Sparsification strategy

- Allow entries whose magnitude exceeds a "drop tolerance"

$$\widehat{s}_{k\ell} = \begin{cases} \bar{s}_{k\ell} & \text{if} \quad \bar{s}_{k\ell} \geq \epsilon(|\bar{s}_{kk}| + |\bar{s}_{\ell\ell}|) \\ 0 & \text{else} \end{cases}$$

Two-level preconditioner [Carvalho, Giraud, Le Tallec, 01]



- Domain based coarse space correction
- $M = M_{AS} + R_O^T A_O^{-1} R_0$ where $A_0 = R_0 S R_O^T$

Outline

Computational framework

Target computer

- IBM-SP4 @ CERFACS (216 procs)
- Blue Gene @ CERFACS (2048 procs)
- System X @ VIRGINIA TECH (2200 procs)

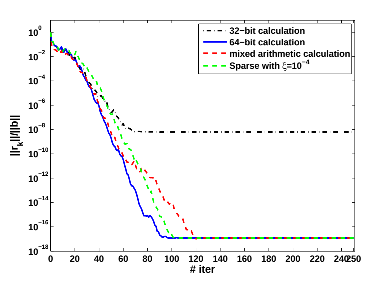
Local direct solver : MUMPS [Amestoy, Duff, Koster, L'Excellent - 01]

- Main features
  - Parallel distributed multifrontal solver (F90, MPI)
  - Symmetric and Unsymmetric factorizations
  - Element entry matrices, distributed matrices
  - Efficient Schur complement calculation
  - Iterative refinement and backward error analysis
- Public domain: new version 4.7.3
  `www.enseeiht.fr/apo/MUMPS - mumps@cerfacs.fr`

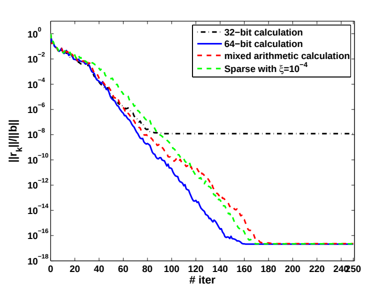
Backward error convergence history

- Mixed arithmetic and Sparse preconditioners behavior

3D Poisson problem



3D Heterogenous diffusion

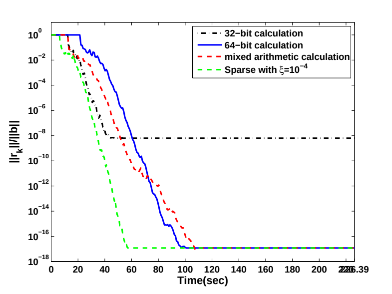


- Convergence history on the Schur complement for a problem with 43 millions dof mapped on 1000 processors

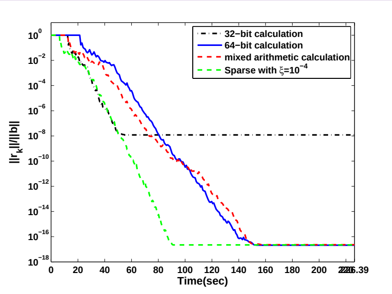
Backward error history vs time

- Mixed arithmetic and Sparse preconditioners behavior

3D Poisson problem



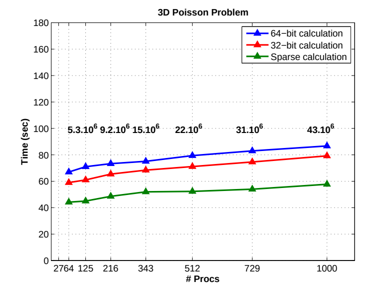
3D Heterogenous diffusion



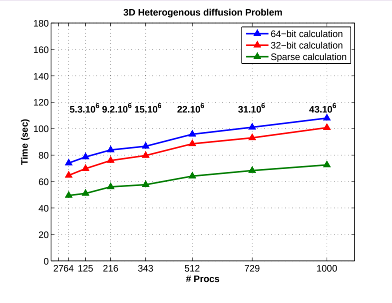
- Time (setup preconditioner + iterative loop) history on the Schur complement for a problem with 43 millions dof mapped on 1000 processors

Numerical scalability in *3D*

3D Poisson problem



3D Heterogenous diffusion

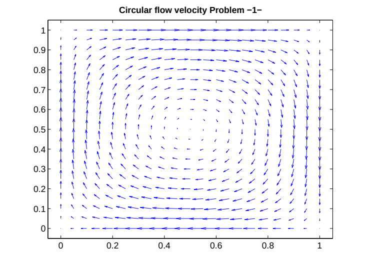


- The trend is similar for all variants using CG Krylov slover
- The computing time increases slightly when increasing # sub-domains
- The solved problem size vary from 1.1 up to 43 Millions of unknowns

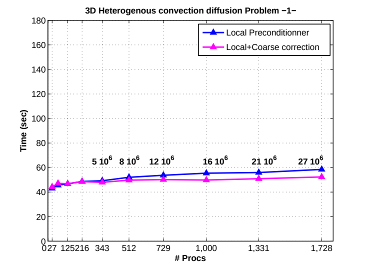
Numerical scalability in $3D$

$$-div(K.\nabla\phi) + v.\nabla\phi = f$$

xy plan view of the circular velocity field



Heterogenous convection diffusion

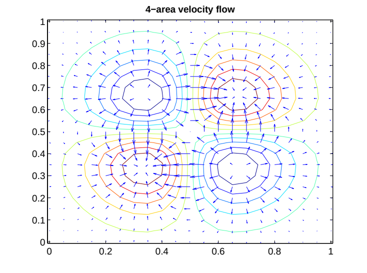


- The trend is similar for all variants using GMRES Krylov slover
- The computing time increases slightly when increasing # sub-domains

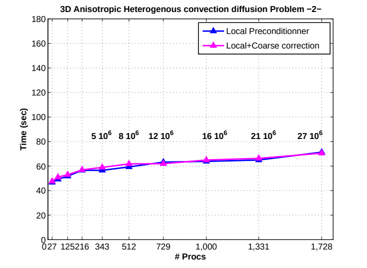
Numerical scalability in 3*D*

$$-div(K.\nabla\phi) + v.\nabla\phi = f$$

xy plan view of the 4-area velocity field



Anisotropic Heterogenous convection diffusion

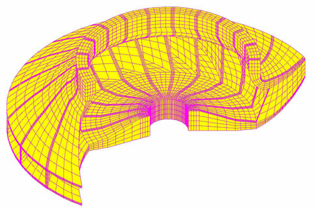


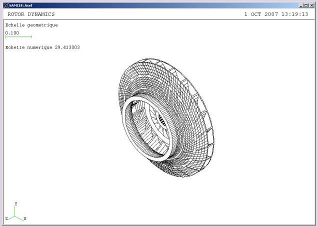
- The trend is similar for all variants using GMRES Krylov slover
- The computing time increases slightly when increasing # sub-domains

Outline

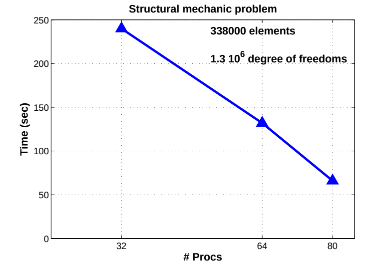
Perspectives joint work with S. Pralet, SAMTECH

Structural mechanic problem





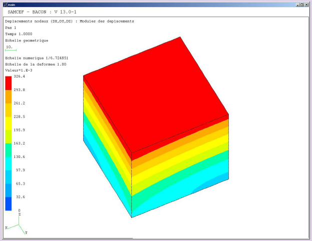
Parallel Performance and Scalability

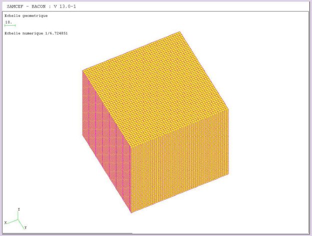


# processors	32	64	80
# iter	108	160	170
Total time(sec)	240	132	66

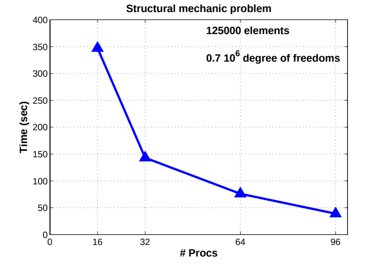
Perspectives joint work with S. Pralet, SAMTECH

Structural mechanic problem





Parallel Performance and Scalability



# processors	16	32	64	80
# iter	40	50	69	73
Total time(sec)	347	143	76	39

Perspectives

Objective

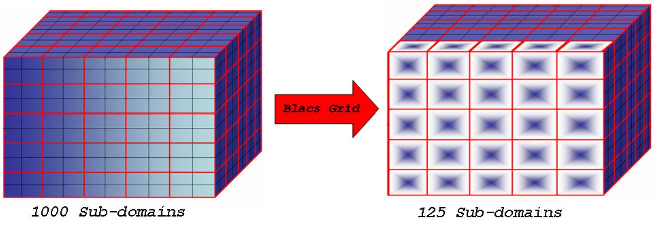
- Study the behavior of the preconditioner on more general industrial problems
- Control the growth of iterations when increasing the # processors

Various possibilities

- Numerical remedy: two-level preconditioner
  - Coarse space correction, ie solve a closed problem on a coarse space
  - Various choices for the coarse component (eg one d.o.f. per sub-domain)
- Computer Science remedy : several processors per sub-domain
  - Two-level of parallelism
  - 2D cyclic data storage

Parallel computing alternative

Main characteristics of the two-level of parallelism



- Allocate each subdomain to many processors
- Benefit from the parallel efficiency of direct solver

More details

Acknowledgements

- More details on this work can be found in [1, 2, 3, 4].
- This research activity was partially developed in the framework of the ANR-CIS project Solstice (ANR-06-CIS6- 010).

THANKS

Bibliography

L. M. Carvalho, L. Giraud, and G. Meurant.
Local preconditioners for two-level non-overlapping domain decomposition methods.
Numerical Linear Algebra with Applications, 8(4):207–227, 2001.

L. Giraud, A. Haidar, and L. T. Watson.
Parallel scalability study of three dimensional additive Schwarz preconditioners in non-overlapping domain decomposition.
Technical Report TR/PA/07/05, CERFACS, Toulouse, France, 2007.
Also appeared as ENSEEIHT-IRIT Technical report RT/APO/07/01.
Under revision for Parallel Computing

L. Giraud, A. Haidar, and L. T. Watson.
Mixed-precision preconditioners in parallel domain decomposition solvers.
Technical Report TR/PA/06/84, CERFACS, Toulouse, France, 2006.
Also appeared as ENSEEIHT-IRIT Technical report RT/APO/06/08.
Accepted for publication in the proceedings of DD17.

J. Langou, J. Langou, P. Luszczek, J. Kurzak, A. Buttari, and J. Dongarra.
Exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy.
Technical Report LAPACK Working Note #175 UT-CS-06-574, University of Tennessee Computer Science, April 2006.