A Parallel Implementation with a comparative study of GMRES(m) Preconditionned by Multiplicative Schwarz, Additive Schwartz and Schur Complement

E. Kamgnia (2) G. A. Atenekeng Kahou (1) M. SoSonkina (3) B. Philippe (1)

(1) http://www.irisa.fr/activites/equipes/sage (gaatenek, philippe) @irisa.fr ,

(2) kamgnia@uycdc.uninet.cm (3) masha@scl.ameslab.gov

(1) Inria/Irisa, Rennes (2) University of Yaounde I, Yaound , Cameroun

(3) Ames Laboratory, Iowa State University, Ames, IA 50011 USA

Outlines

- Parallelizing GMRES(m) with
  - Mutiplicative Schwarz preconditioning
    - Multiplicative Schwarz preconditioning
    - Parallel building of Krylov basis
  - pARMS description
    - Additive Schwarz preconditioning
    - Distributed Schur Complement preconditioning
- Comparative results

1. Parallelizing GMRES(m)

IRISA

GMRES(m) builds an orthonormal basis V_{m+1} of $\mathcal{K}_{m+1}(A, r^k)$ (QR factorization, usually obtained by the Modified Gram-Schmidt process).

and exploits the property

$$y = V_m z \text{ minimizes } \|b - A(x_0 + y)\|$$

$$\text{iff} \quad z \text{ minimizes } \| \|r^k\| e_1 - \bar{H}_m z\|_2$$

where $\bar{H}_m = {V_{m+1}}^T A V_m$ (Hessenberg matrix) to generate the next iterate $x_{k+1} = x_k + y$

Therefore : the computational kernel of each GMRES(m) iteration has two main procedures :

- $x \longrightarrow Ax$: matrix x vector multiplication (m times),
- QR factorization (MGS),

and a nonexpensive linear least square problem.

Parallelism in GMRES(m) (cont.)

IRISA

General remark on GMRES(m) : The classical version implements MGS on $[r^k, Av_1, Av_2, \cdots, Av_m]$.

Which implies recursion : v_{l+1} can only be computed when v_l is known.

Therefore, parallelism can only be expressed in

- the multiplications by the preconditionned sparse matrix (Parallelism is efficient when the matrix is stored but it might be poor for a preconditioned matrix),
- the vector operations (BLAS 1) of the orthogonalizing process.

Unfortunately, inner products are not scalable and potential parallelism is limited.

Decomposition with overlapping blocks

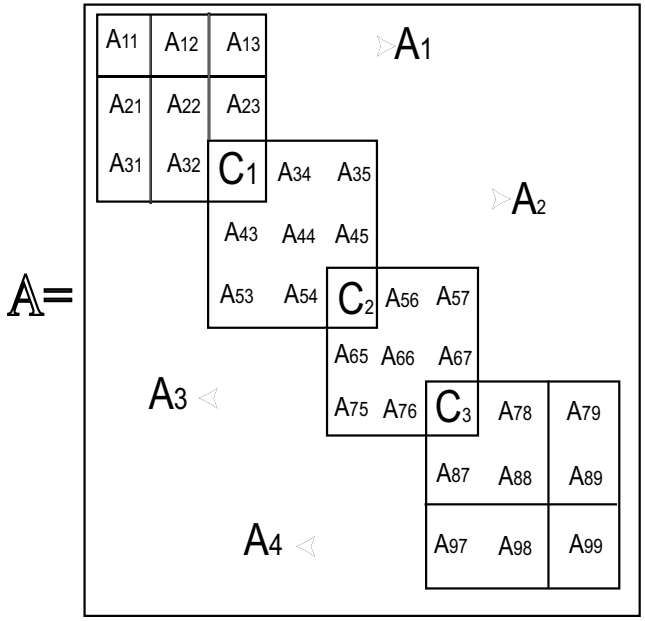
IRISA

Given an example of block structure as follow:

$$A = \begin{pmatrix} A_{54} & & A_{11} & & A_{13} & \\ & A_{12} & & & & \\ A_{67} & & A_{75} & & & \\ & & & A_{53} & & A_{77} \\ A_{21} & & A_{76} & & & \\ & & & A_{56} & & A_{33} \\ & & A_{35} & & & \\ A_{22} & & & & & A_{34} \\ & A_{65} & & & A_{55} & \\ & & A_{57} & & & A_{78} \\ A_{66} & & A_{43} & A_{79} & & A_{44} \\ & & & A_{32} & & \\ A_{23} & & A_{31} & & A_{43} & \\ & & & & & A_{45} \end{pmatrix}$$

Decomposition with overlapping (D.W.O)bloc

Decomposition with overlapping is obtained by reordering:



The decomposition can be obtained :

- from the PDE domain,
- “by hand” after a bandwidth reduction,
- from a dedicated tool (see an upcomming talk).

Decomposing matrix A into 4 overlapping blocks.

Multiplicative Schwarz Method

IRISA

- The so-called additive and multiplicative Schwarz methods, are iterative methods which combine at every step local solves on each domain. The methods are relaxation methods corresponding to two splittings of the matrix.
- From an algebraic point of view, these methods converge when the matrix is SPD or M-matrix [see Frommer and al.]
- Due to a frequent low rate of convergence, the methods are preferably used as preconditionners for Krylov methods.

Notations

IRISA

Notation :

- x^k : current iterate ;
- $r^k = b - Ax^k$ corresponding residual;
- $\bar{A}_i$ is A_i augmented by identity to the dimension of A;
- $\bar{C}_i$ is C_i augmented by identity to the dimension of A;
- A_i^+ is A_i^{-1} augmented with zeros to the dimension of A;

Data: A_i, x^i, r^k and A

Result: $x^{k+1} = x$,$r^{k+1} = r$

for $i = 1 : p$ **do**

$x := x + A_i^+ r$;

$r := r - AA_i^+ r$;

end

Multiplicative Schwarz iteration

IRISA

It follows that

$$r^{k+1} = (I - AA_p^+) \cdots (I - AA_1^+) r^k.$$

$\Rightarrow$ relaxation method defined by some splitting $A = M - N$, iteration matrices for the residual and error are respectively

$$\begin{aligned} NM^{-1} &= (I - AA_p^+) \cdots (I - AA_1^+), \\ M^{-1}N &= (I - A_p^+ A) \cdots (I - A_1^+ A) \end{aligned}$$

From $N = M - A$, equivalent expressions for right and left preconditioning are :

$$\begin{aligned} M^{-1}A &= I - (I - A_p^+ A) \cdots (I - A_1^+ A), \\ AM^{-1} &= I - (I - AA_p^+) \cdots (I - AA_1^+). \end{aligned}$$

Explicit formulation

IRISA

We have proved that M^{-1} can be expressed by :

$$M^{-1} = \bar{A}_p{}^{-1}\bar{C}_{p-1}\bar{A}_{p-1}^{-1}\bar{C}_{p-2}\cdots\bar{A}_2{}^{-1}\bar{C}_1\bar{A}_1{}^{-1}$$

- Advantages of this explicit formulation
  - Calculations of x_k and of r_k are separated. Since the calculation of r_k is obtained by a matrix$\times$vector multiplication, it can be parallelized.
  - The cost of the classical expression is marginaly higher than the cost of the explicit formulation.

A priori construction of the Krylov basis

Traditionally, one constructs an orthogonal basis of

$$\mathcal{K}_m(AM^{-1}, v) = span\{v, (AM^{-1})v, \cdots, \prod_{j=1}^{m-1} (AM^{-1})v\};$$

which is ill conditioned; instead use following space:

$$\bar{\mathcal{K}}_m(AM^{-1}, v) = span\{\sigma_0 v, \sigma_1(AM^{-1} - \lambda_1 I)v, \cdots, \sigma_{m-1} \prod_{j=1}^{m-1} (AM^{-1} - \lambda_j I)v$$

where $\{\sigma_i\}$, $\{\lambda_j\}$ are constants chosen for better conditioning. The construction of the basis proceeds in two stages :

A priori construction of the Krylov basis Cont

A non orthonormal basis $[Y_{m+1}]$ of $\mathcal{K}_{m+1}(AM^{-1}, r)$ is built from $y_1 = r$ and by $k = 1, \cdots, m$

$$y_{k+1} = \mu_k(AM^{-1}y_k - \sigma_k y_k).$$

where (μ_k) and (σ_k) are predefined sets of constants. Therefore

$$AY_m = Y_{m+1}\bar{T}_m,$$

where $\bar{T}_k \in^{(k+1)\times k}$ is the bidiagonal matrix

$$\bar{T}_m = \begin{pmatrix} \sigma_1 & & \\ \frac{1}{\mu_1} & \ddots & \\ & \ddots & \sigma_m \\ & & \frac{1}{\mu_m} \end{pmatrix}.$$

Recovering Arnoldi's relation

IRISA

From the QR factorisation $Y_{m+1} = V_{m+1} R_{m+1}$,
the basis V_{m+1} of $\mathcal{K}_{m+1}(AM^{-1}, r)$ is orthonormal and the Arnoldi's relation

$$AV_m \quad = \quad V_{m+1} \bar{H}_m$$

can be recovered by $\bar{H}_m = R_{m+1} \bar{T}_m R_m^{-1}$.
Since $cond_2(Y_m) = cond_2(R_m)$, basis Y_m must not be too ill-conditioned.
There exist strategies in selecting appropriate (μ_k) and (σ_k).
[Calvetti-Petersen-Reichel 93]
Parallel EXPOVEC [Sidje94], Parallel GMRES [Erhel94].

Pipelining the basis construction

IRISA

with the preconditioned operator

$$
\begin{array}{cccccc}
 & & & \rightarrow & & \\
 & & & y_1 & y_2 \quad \cdots & y_m \\
 & & & \times & \times \quad \times & \times \\
\downarrow & A_1 & \times & . & . \quad . & . \\
 & A_2 & \times & . & . \quad . & \\
 & \vdots & \times & . & . \quad \searrow & \\
 & A_p & \times & . & &
\end{array}
$$

Horizontal direction : $y_{i+1} = \mu_i(AM^{-1} - \sigma_i I)y_i$ for $i = 1 : m$.

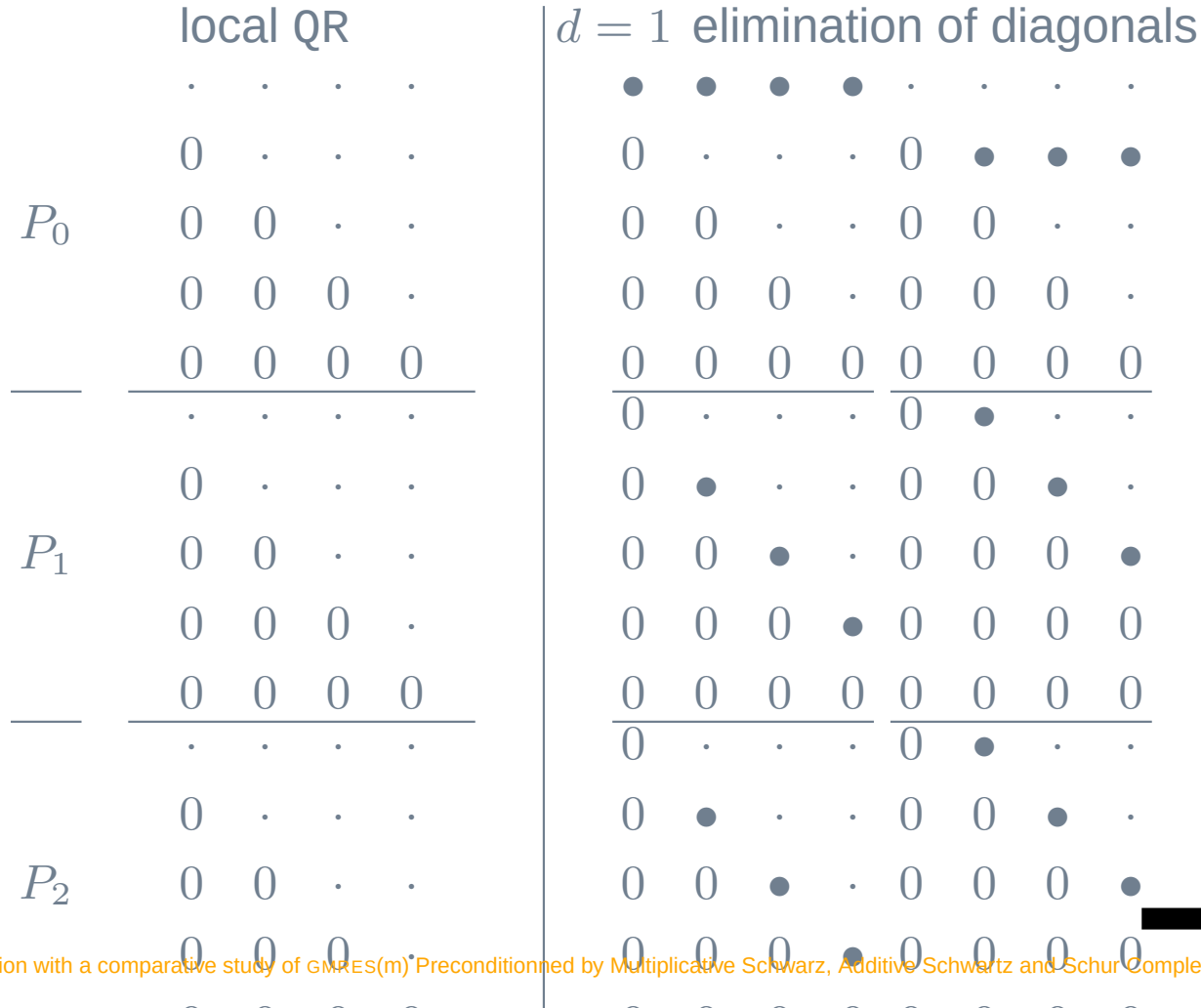
Vertical direction : recursion through the blocks. Dependence analysis proves that efficiency is equal to 1/3 (1/2 when weak overlap). See the profiling (I. Souopgui)

Parallel QR factorisation

IRISA

Parallel QR which does not involve scalar products [Sidje 94].

Two steps : local QR + eliminations of all, but the first, triangles.



pARMS Description

IRISA

- Solution using methods from pARMS package [Li, Saad, Sosonkina 2000].
- Features of pARMS
  - matrix row based partitioning
  - no data redistribution
  - no collection of data on a single node or a group of (nodes)
  - no data replication except for overlapping

pARMS Description Cont....

IRISA

- Based on Algebraic Recursive Multilevel Solver [Saad 1999]
- Several types of unknowns are distinguished and handled differently to construct a preconditioner
- Preconditioners tested

  add_arms Additive Schwarz procedure, with local ARMS on sub-domains, with and without overlap.

  lsch_arms Schur complement technique, with ARMS as factorization used for local submatrix. Equivalent to Additive Schwarz preconditioner on Schur complement.

Distributed Schur Complement Con

- In each processor:

$$\begin{pmatrix} u_i \\ y_i \end{pmatrix} \quad \begin{matrix} \text{Internal variables} \\ \text{Interface variables} \end{matrix}$$

- Eliminate u_i from the system of local equations:

$$S_i y_i + \sum_{j \in N_i} E_{ij} y_j = g_i - E_i B_i f_i \equiv g_i',$$

where S_i is the "local" Schur complement
$S_i = C_i - E_i B_i F_i$.

- Schur complement system:

$$\begin{pmatrix} S_1 & E_{12} & \dots & E_{1p} \\ E_{21} & S_2 & \dots & E_{2p} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} g_1' \\ g_2' \end{pmatrix}$$

Distributed Schur Complement

IRISA

- Solve reduced (Schur Complement) system approximately.
- With distributed data structure, global Schur Complement matrix needs no explicit construction:
  - No extra communications, storage, and assembly
- Block Jacobi preconditioner for global Schur Complement system [Saad, Sosonkina 1999].
  - using approximate inverses
  - Derived via (2×2) block LU factorization of the original matrix:
- Locally, this LU factorizations is given as

$$\text{If} \quad A_i = \begin{pmatrix} L_{B_i} & 0 \\ E_i U_{B_i} & L_{S_i} \end{pmatrix} \begin{pmatrix} U_{B_i} & L_{B_i} F_i \\ 0 & U_{S_i} \end{pmatrix} \quad \text{Then} \quad L_{S_i} U_{S_i} = S_i$$

Experiments

Architecture :

Network of nodes connected through a Giga-bit Ethernet switch.

Each node : bi-processor Sun Fire V20z (Opteron at 2.2 GHz).

Test matrices :

Five from Device simulation problems (from O. Shrenke)

Method : GMRES preconditioned by one step of MS on the right side.

Direct method for the local solves : SUPERLU.

Experimentations: description

- Systems scaled by 1-norm;
- Zero initial guess
- Artificial RHS b, such that Ae=b was used.
- Legend for the tables:
  - proc – number of processors
  - n – system size
  - pr – preconditioner type:
    - 21 – Multiplicatve Schwarz;
    - 20 – Additive Schwarz non-overlapping;
    - 2 – Additive Schwarz with one level of overlapping;
    - 1 – Schur Complement.

Experimentations: description

IRISA

- nnz –nonzero ratio used in preconditioner:
- iou –number of outer iterations (maximum number was given as 700)
- iin –number of inner iterations
- tto –total time for solution, seconds, (excludes partitioning)
- tpi –time per outer iteration
- tpr –time to construct preconditioner
- residual –achieved residual

Experimentations

IRISA

igbt3 matrix

proc	n	pr	nnz	iou	iin	tto	tpi	tpr	residual
4	10938	2	0.40	114	561	3.73	0.0327	0.31	1.286e-14
4	10938	1	0.39	164	825	1.87	0.0114	0.27	8.405e-15
4	10938	20	0.39	236	1176	6.58	0.0279	0.27	1.237e-14
4	10938	21		4	80	1.57	0.018	0.11	8.721e-16
8	10938	1	0.20	180	905	1.41	0.0078	0.15	1.548e-14
8	10938	2	0.20	397	718	7.85	0.0198	0.18	1.406e-14
8	10938	20	0.20	700	1373	10.43	0.0149	0.15	1.5e-09
8	10938	21		68	1908	19.79	0.010	0.05	9.96e-16
32	10938	1	0.05	397	1990	2.29	0.0058	0.03	1.416e-14
32	10938	20	0.05	700	1000	2.45	0.0035	0.03	5.313e-06
32	10938	2	0.05	700	1055	3.44	0.0049	0.04	8.431e-12
32	10938	21		700	7000	48.59	0.006	0.01	9.39e-16

Experimentations Cont.....

IRISA

bjtcai matrix

proc	n	pr	nnz	iou	iin	tto	tpi	tpr	residual
4	27628	1	0.51	296	1485	8.82	0.0298	0.87	2.687e-14
4	27628	2	0.51	700	2387	60.18	0.0860	0.92	3.004e-13
4	27628	20	0.51	700	3148	70.65	0.1009	0.87	6.368e-08
4	27628	21		41	656	21.40		0.27	6.159e-16
8	27628	1	0.25	300	1505	4.53	0.0151	0.43	2.334e-14
8	27628	2	0.25	700	1438	24.87	0.0355	0.48	4.636e-12
8	27628	20	0.25	700	2041	24.92	0.0356	0.42	9.684e-09
8	27628	21		55	830	19.44			8.707e-16
32	27628	1	0.06	495	2480	4.19	0.0085	0.09	1.956e-14
32	27628	2	0.06	700	1356	8.06	0.0115	0.11	9.604e-07
32	27628	20	0.06	700	1389	6.65	0.0095	0.08	4.201e-05
32	27628	21		344	5609	73.78	0.0181		9.70e-16

Experimentations Cont....

IRISA

matrix9 matrix

proc	n	pr	nnz	iou	iin	tto	tpi	tpr	residual
4	103430	2	0.78	109	491	140.18	1.2861	6.61	3.312e-14
4	103430	20	0.78	388	1587	351.84	0.9068	5.94	2.293e-14
4	103430	1	0.86	85	430	30.58	0.3598	5.94	2.493e-14
4	103430	21		2	66	150.12	29.20	91.63	3.715e-16
8	103430	2	0.48	166	486	101.48	0.6113	3.57	2.332e-14
8	103430	20	0.41	568	1313	211.05	0.3716	2.82	2.494e-14
8	103430	1	0.41	89	450	15.35	0.1725	2.82	3.156e-14
8	103430	21		2	66	39.01	0.3598	25.73	2.75e-16
16	103430	2	0.21	207	719	59.38	0.2869	1.92	2.604e-14
16	103430	20	0.21	528	1646	85.09	0.1612	1.50	2.437e-14
16	103430	1	0.25	94	475	8.53	0.0907	1.50	3.835e-14
16	103430	21		2	67	39.01	0.33	16.88	5.32e-16

Experimentations Cont.....

IRISA

new3 matrix

proc	n	pr	nnz	iou	iin	tto	tpi	tpr	residual
4	125329	1	0.52	163	819	49.78	0.3054	5.67	2.815e-14
4	125329	2	0.57	277	723	322.64	1.1648	9.57	4.592e-14
4	125329	20	0.52	700	2352	598.44	0.8549	5.67	8.446e-11
4	125329	21		6	169	90.48	0.39	22.98	1.85e-16
8	125329	1	0.26	298	1495	42.73	0.1434	2.88	4.097e-14
8	125329	2	0.29	488	1188	339.27	0.6952	5.13	3.597e-14
8	125329	20	0.26	700	1903	260.81	0.3726	2.87	1.061e-06
8	125329	21		12	299	55.75	0.3726	7.642	8.52e-16
16	125329	1	0.13	204	1025	16.50	0.0809	1.23	2.984e-14
16	125329	20	0.13	700	1300	98.68	0.1410	1.23	8.117e-06
16	125329	2	0.15	700	941	184.99	0.2643	3.38	9.461e-10
16	125329	21		52	676	92.61	0.133	2.58	8.486e-16

Experimentations

IRISA

Usound.out

proc	n	pr	iou	nnz	tto	tpi	tpr	residual
4	531441	20	29	0.36	175.32	6.0455	65.60	3.529e-13
4	531441	2	24	0.46	232.16	9.6733	86.22	1.962e-13
4	531441	1	23	0.36	121.83	5.2970	65.76	2.756e-13
4	531441	21	3	99	1002.18	5.2970	2.63	4.53e-16
8	531441	20	29	0.18	92.83	3.2010	29.68	3.577e-13
8	531441	2	23	0.24	128.75	5.5978	47.34	2.738e-13
8	531441	1	22	0.18	58.96	2.6800	29.79	2.808e-13
8	531441	21	4	132	912.18	1.62		3.43e-16
16	531441	20	31	0.08	51.07	1.6474	15.77	3.724e-13
16	531441	2	23	0.14	84.36	3.6678	26.74	2.46e-13
16	531441	1	22	0.08	32.51	1.4777	15.99	2.586e-13
16	531441	21	4	112	744.60	68.53		3.40e-16

Conclusions

- The MS has a better performance in terms of number of iterations
- SC is better in terms of scability