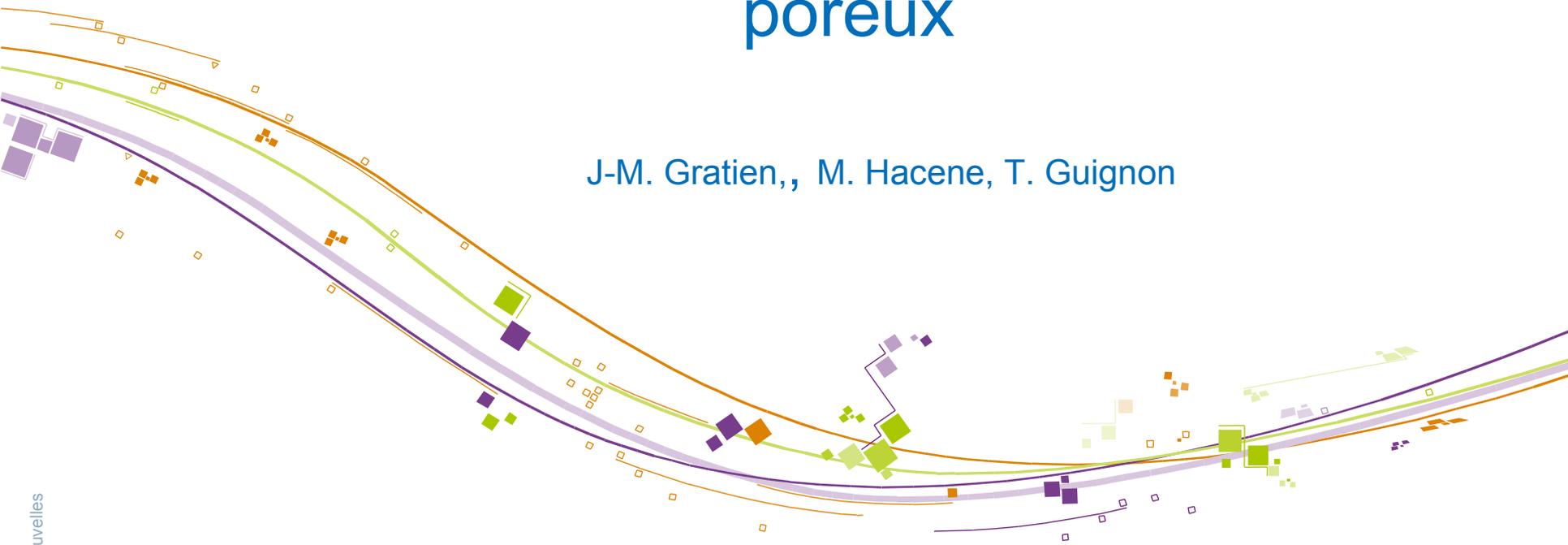
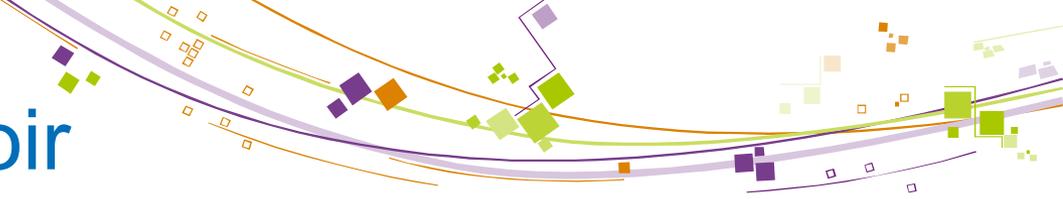


Solveurs linéaires sur GPU pour la simulation d'écoulement en milieux poreux

J-M. Gratien,, M. Hacene, T. Guignon

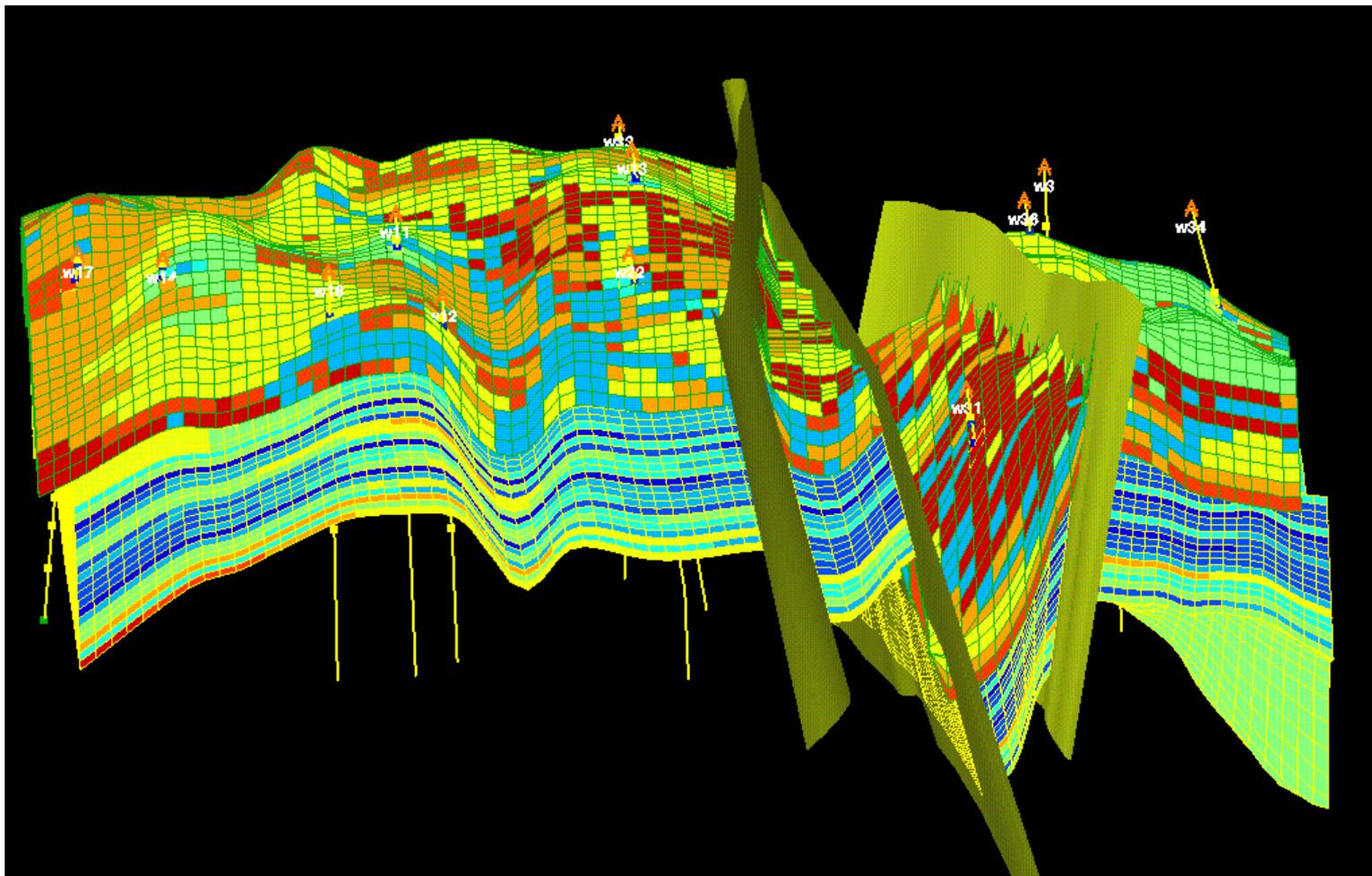


Simulateur de réservoir



- **Réservoir géologique:**
 - Milieux poreux contenant eau huile et gaz.
 - Puits producteurs, puits injecteurs
- **Objectif de la simulation :**
 - Prévision de la production
 - Étudier les scénario de développement
- **Écroulement multiphasique en milieux poreux:**
 - Loi de darcy et Thermodynamique.
 - Schéma volumes finis.
 - Grille non structurée : failles, mailles « mortes » disparition de couches (pinch out) , LGR .

Grille réservoir



Simulation et solveur linéaire.

- **A chaque pas de temps : système non linéaire :**
 - Newton.
 - À chaque pas de Newton : résolution de $Ax = b$.
 - 80 % du temps passé dans le solveur.
- **A: système linéaire creux non structuré :**
 - Non symétrique
 - Format CSR bloc (3x3 cas Black Oil 3 phases)
 - Graphe adjacence correspond à la connectivité de la grille.
 - Équations puits.
- **Méthode “classique” : Krylov pré-conditionné**
 - BiCGStab.
 - ILU0 ou CPR-AMG.

Solveur linéaire sur GPU

■ Objectif :

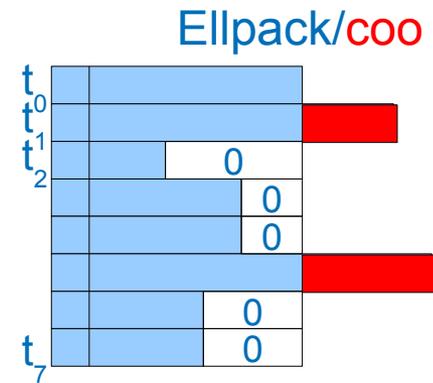
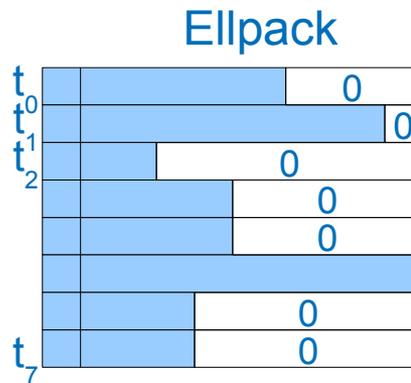
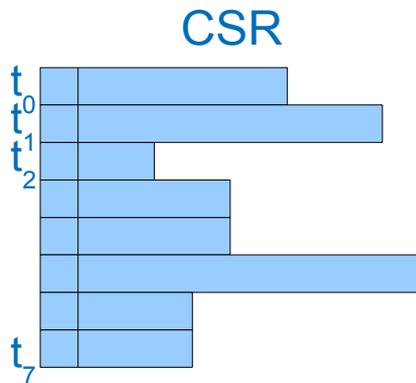
- Effectuer la résolution itérative sur GPU
- Conserver le calcul du preconditionneur sur CPU (ILU0 : factorisation)

■ Composants logiciels sur GPU :

- Produit matrice creuse vecteur (spmv) $y = A.x$
- Application du preconditionneur (ILU0: résoudre $L.U.x = y$)
- Algèbre linéaire (BLAS 1) : Nvidia CUBLAS

Spmv sur GPU

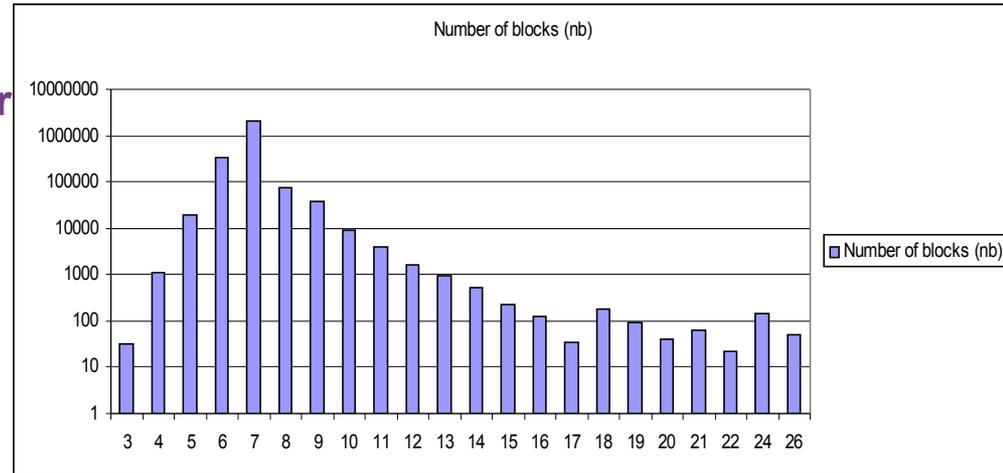
- **Contrainte GPU modèle d'exécution // : SIMD thread**
 - Tous les threads (dans un « groupe ») doivent faire la même chose au même moment.
 - Grain fin : un thread (ou +) pour chaque ligne de A.
- **csr spmv non efficace:**
 - Largeur de ligne variable
 - threads inactifs sur les lignes plus courtes.



Spmv IFP

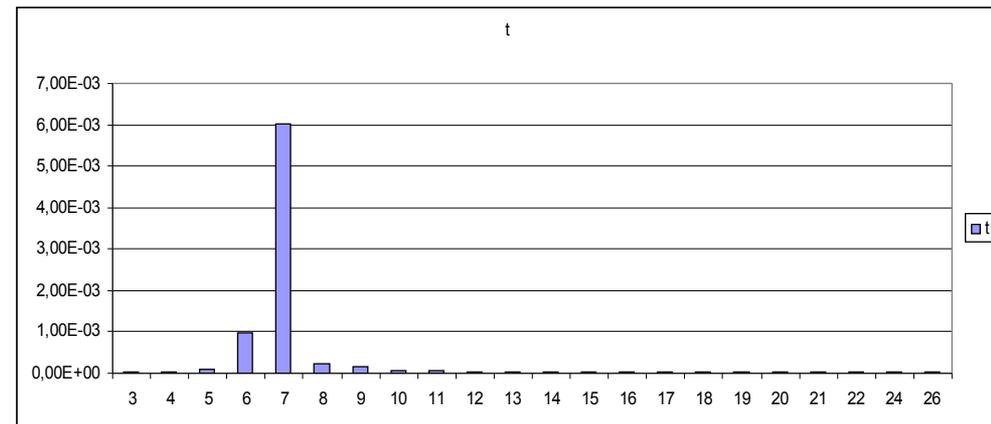
■ Solution proposée:

- Grouper les lignes de même largeur (renuméroter les équations).
- 1 "Kernel" GPU différent pour chaque largeur de ligne.
- Calcul $y = P.A.P^{-1}.x$ ou $y = P.A$ au lieu de $y = A.x$



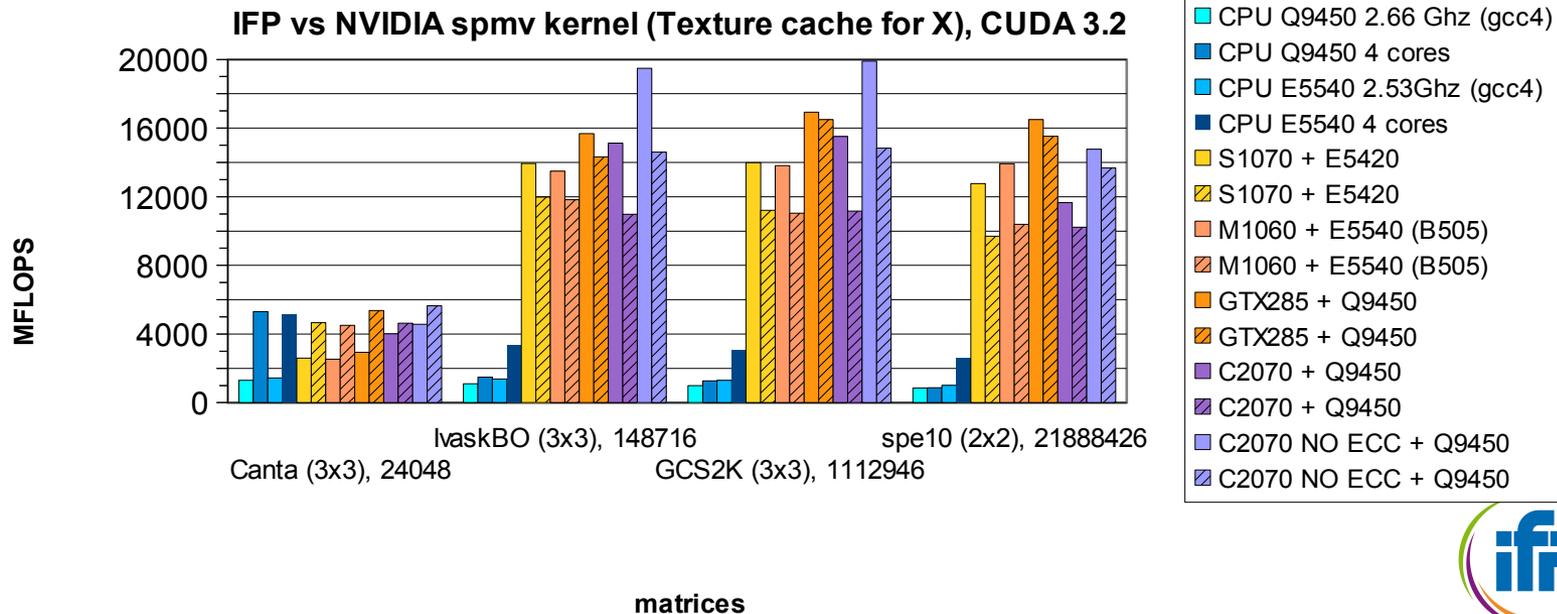
■ Cas IFP : système linéaire extrait du simulateur

- Majorité de lignes de largeur 7 et 6



Spmv: performances (dp)

- Comparé à un Intel Nehalem (E5540 @2.53Ghz)
 - GPU/ 1 core acc: x10 (grand système) jusqu'à x15
 - GPU/ 4 cores acc: au moins x4 (grands systèmes)
- Multi coeur: pas d'extensibilité // au delà de 2 cœurs (grand systèmes):
 - E5540 4 cores : x2.5 au mieux
- Effet de cache (L3) favorable au CPU pour les petits systèmes



ILU0 sur GPU

- $A \approx L.U$ sur CPU.
- Résolution $L.U.x = y$ sur GPU (systèmes triangulaires):
 - Ordre naturel de la matrice ne permet pas un degré de parallélisme élevé.
 - Une résolution massivement // de systèmes triangulaires nécessite une permutation des équations/inconnues.
- Coloriage du graphe de la matrice :
 - Attribuer à chaque nœud une couleur différentes des nœuds voisins.
Ordonner la matrice en groupant les équations (nœuds) de même couleur
- Réordonner la matrice diminue l'efficacité de ILU0.

ILU0 sur GPU

- L_i, U_i : blocs creux

- résolution $L.x = y$:

- $x_1 = D_1^{-1} \cdot y_1$

- $x_2 = D_2^{-1} \cdot y_2 - L_2 \cdot x_1$

- résolution $U.x = y$:

- $x_4 = D_4^{-1} \cdot y_4$

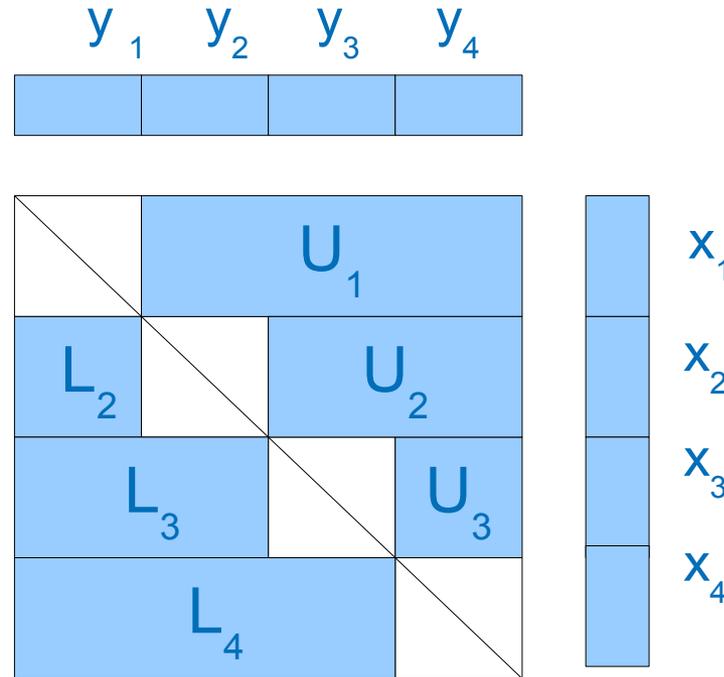
- $x_3 = D_3^{-1} \cdot y_3 - U_3 \cdot x_4$

- GPU :

- Résolution système diagonal.

- Spmv : Nvidia hybrid.

- IFPEN spmv : permutation des vecteurs solution

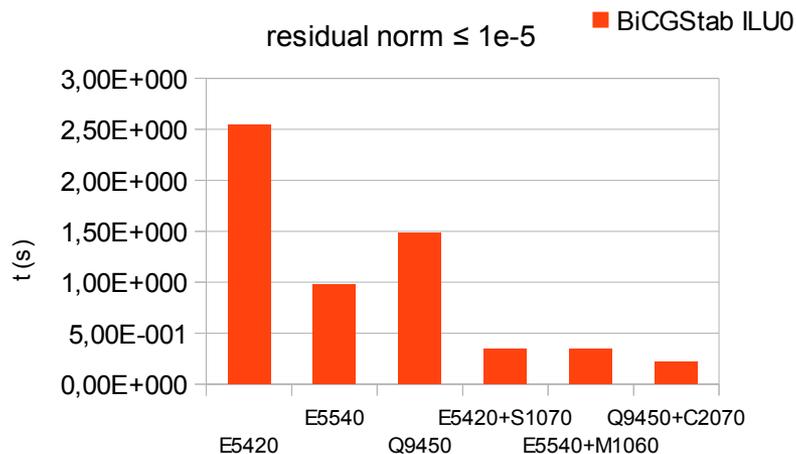


Preconditionneur polynomial

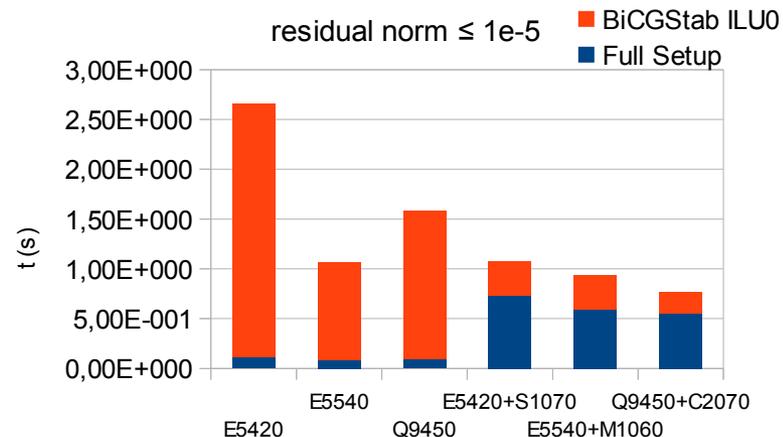
- **Polynôme de Neuman:**
 - $I+N+N^2+N^3+\dots+N^d$ with $N=I-w.D^{-1}.A$
 - Approx de la plus grande valeur propre (puissance itérée).
- **Basée sur $y=A.x$ et BLAS 1:**
 - Portage immédiat sur GPU.
- **Sur CPU: polynomial non compétitif face à ILU0 :**
 - D : degré du polynôme: d spmv pour chaque application de preconditionneur
 - L.U.x=y équivalent à 1 spmv
- **IFP spmv: $y = P.A.P^{-1}$**

Resultats ILU0

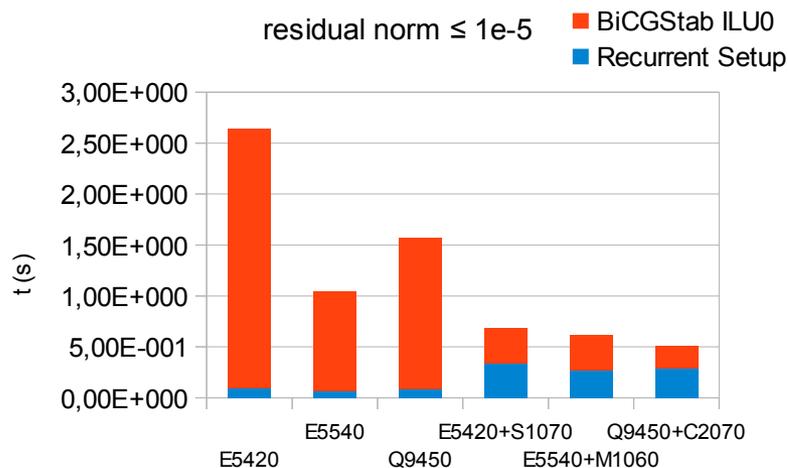
IvaskBO, BiCGStab ILU0 solve



IvaskBO BiCGStab ILU0 solve + full setup



IvaskBO BiCGStab ILU0 solve + recurrent setup



Resultats ILU0 et Polynomial

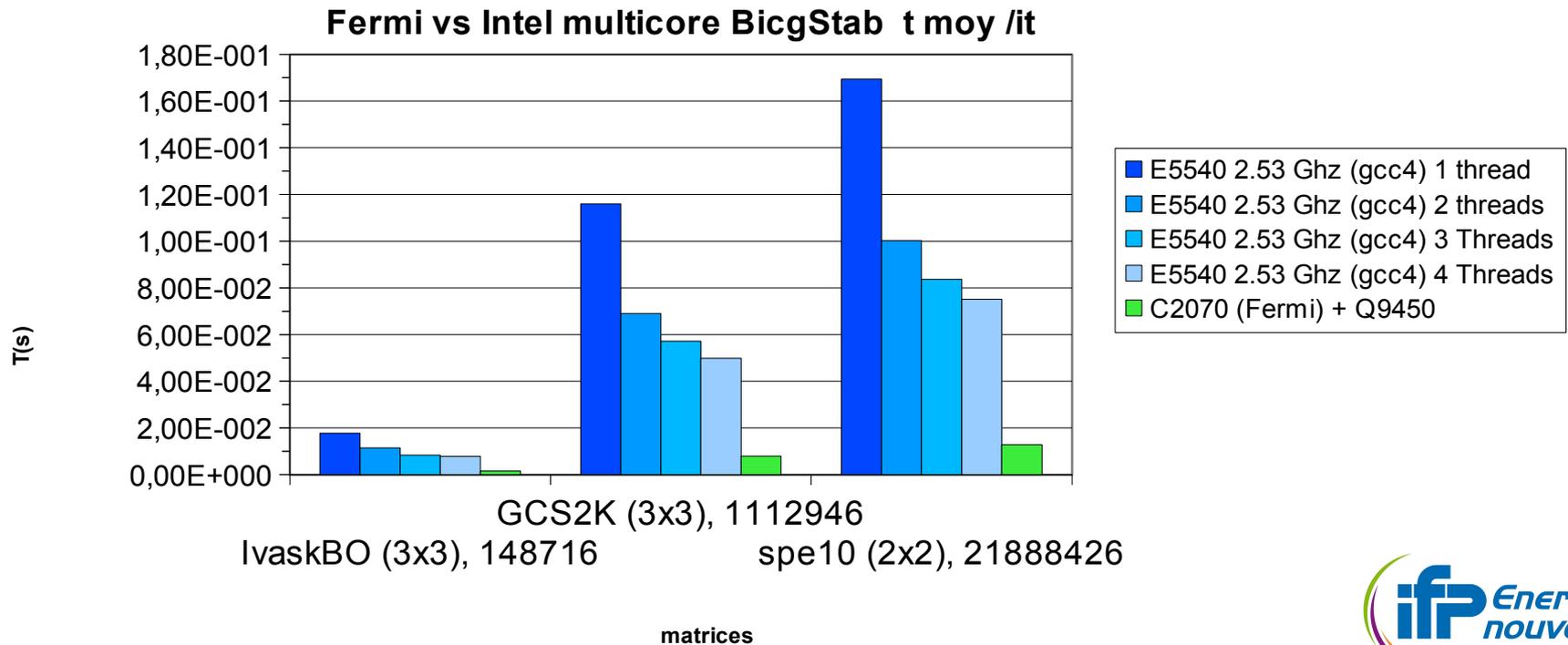
	E5540 vs Q9450 + C2070 acceleration			it CPU/it GPU		
	IvaskBO	GCS2 K	spe10	IvaskBO	GCS2K	spe10
ILU0 Solve	4,51	10,37	2,94	26 / 53	19 / 22	201 / 766
Setup + Solve	1,38	1,64	2,57			
Recurrent setup + solve	2,06	2,95	2,76			
Poly prec (d=2) vs ILU0	3,56	7,81	x	26 / 66	19 / 26	x
Poly prec (d=2) vs ILU0 +setup	2,8	4,05	x			
Poly prec (d=2) vs ILU0 + recurrent setup	2,76	3,96	x			

Équilibre GPUs/cœurs CPU.

- **cores than GPU**
 - B505 blade (Bull) : 2 cpus (4 coeurs) 2 GPUs : 1 GPU / 4 c.
 - Station avec C2070 : 1 cpu (4 coeurs) 1 GPU : 1 GPU / 4 c.
 - Noeud bi CPU avec S1070 : 2 cpus (4 coeurs) 4 GPU : 1 GPU / 2 c.
- **Équilibre équitable: 1 GPU contre 4 cœurs CPU.**
- **Actuellement:**
 - On ne dispose pas encore de ILU/Poly multi-coeurs.
- **Comparaison multi-coeurs CPU/GPU avec BicgStab non préconditionné.**

Multi cœurs CPU vs GPU

- **BicgStab 1/2/3/4 cores (E5540) vs Fermi (C2070+Q9450)**
 - Multithread (posix threads), BLAS: Intel MKL, parallélisme à grain moyen.
 - Acc 4 coeurs : de x2.24 à x 2.33
- **E5540 4 cores/Fermi : from x4,9 to x6,3.**
- **Acc L.U.x = y inférieure à acc spmv et BiCGStab**



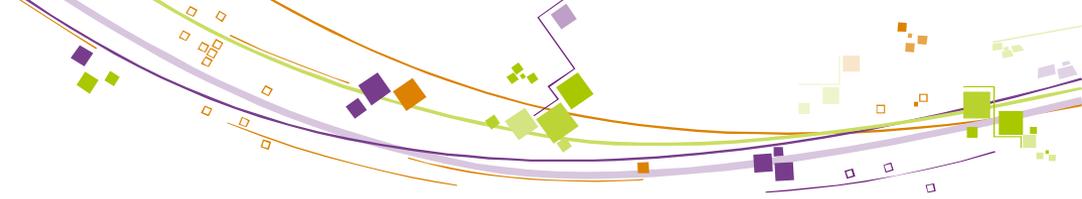
Expériences in situ

- **Simulateur d'injection de CO2 CooresArcane (ANR SHPC02)**
 - Injection de CO2 dans un aquifère.
- **Systeme:**
 - Tesla S1070 vs E5420 (2,5 Ghz)

190x120x5 114000	BSSOR GPU	POLY GPU	IFP CPR- AMG CPU	IFP-ILU0 CPU	PETSC-ILU CPU
t. total	2099,91	3913,87	2963,9	7031,61	15563,5?
t. solveur	1409,98	3214,29	2300,54	6381,34	15527,9
NITER	152483	123037	5069	86353	100159
NSTEP	801	801	801	801	784

Conclusions

- **GPU actuels (fermi) vs CPU multi coeurs actuels (Nehalem)?**
- **ILU0:**
 - **Accélérations limitées car:**
 - Coût de mise en place sur GPU importants.
 - Augmentation du nombre d'itérations
 - Effort principal à faire sur la mise en place mais aussi sur ré-ordonnancement.
 - Factorisation ILU0 sur GPU ?
- **Polynomial:**
 - Mise en place et itérations rapides.
 - Risque de non convergence (ex: spe10).
- **BSSOR et autres (SPAI, Chebychev, ...)**
- **GPU vs CPR-AMG sur CPU ?**
 - CPR AMG sur GPU ?



- **Travail supporté par les projets ANR PARA et PETALH**

- **Questions ?**