

DE LA RECHERCHE À L'INDUSTRIE

cea den

PARALLÉLISATION DE CAST3M EN MÉMOIRE DISTRIBUÉE

Club Castem 2015 | Gauthier Folzan

27 NOVEMBRE 2015

- **Parallélisme intra-opérateur : RESOu**
 - Pthreads et fortran
 - Tâches simultanées à l'intérieur de l'opérateur
- **Parallélisme inter-opérateur : ASSIstant**
 - Pthreads et Esope
 - Exécution d'opérateurs en parallèle
- **Parallélisme inter-programme : COLLaborateur**
 - Exécution de plusieurs programmes en parallèle

- Limite des assistants
 - Goulot d'étranglement sur les fonctions GEMAT
- Nouveau niveau de parallélisme
- Exécution simultanée de plusieurs instances de Cast3m (sur un ou plusieurs processeurs) appelées *collaborateurs*
- Plusieurs instances = plusieurs zones GEMAT
 - Gain possible sur les blocages globaux des fonctions GEMAT même sur 1 machine
- Segments d'une zone mémoire accessibles par un seul collaborateur
- Nécessité de communication entre les collaborateurs
 - Cast3m basée sur les objets → Echange d'objets Cast3m
 - Prise en charge par un nouvel opérateur
- Opérateur de communication : COLL(aborateur)
 - Différentes fonctionnalités définies par un mot-clé :
 - DEBUT / FIN / RANG / NOMBRE / ENVOYER / RECEVOIR

- Choix: utilisation de la librairie MPI
- Bibliothèque de fonctions de passage de messages standardisée
- Fonctions d'informations
 - Nombre de processus
 - Rang du processus
- Fonctions d'envoi/réception
 - Fonctions synchrones / asynchrones
 - Communications globales / point à point
- Implémentation utilisée : OpenMPI
 - D'autres choix sont possibles avec peu de changements dans l'opérateur

UTILISATION DE L'OPÉRATEUR COLL(ABORATEUR)

- **Opérateur de communication : COLL(aborateur)**
 - Différentes fonctionnalités définies par un mot-clé :
 - DEBUT / FIN / RANG / NOMBRE / ENVOYER / RECEVOIR

- **Initialisation: DEBUT**
 - Aucun argument
COLL DEBUT ;
 - Opérateur à lancer avant toute communication
 - Initialisation de l'environnement
 - Allocation des historiques

- Fermeture : FIN
 - Aucun argument
COLL FIN;
 - Opérateur à lancer après les communications
 - Fermeture de l'environnement MPI

- Récupération du numéro de collaborateur : RANG
 - ENT1 = COLL RANG;

- Récupération du nombre de collaborateurs : NOMBRE
 - ENT1 = COLL NOMBRE;

- Envoi d'un message : ENVOYER

- COLL ENVOYER ENT1 OBJ1 OBJ2 ... OBJn;

- ENT1 : destinataire du message

- OBJ_i : objets à envoyer

- Type d'objets:

- Entier

- Réel

- Logique

- Mot

- Champ par point

- Champ par élément

- Modèle

- Rigidité

- Maillage

- Point

- Configuration

- Communication synchrone bloquante point à point

- L'envoi ne commence pas tant que le destinataire n'est pas prêt à recevoir

- L'opérateur ne quitte pas tant que le message n'a pas été envoyé

- Réception d'un message : RECEVOIR
 - OBJ1 OBJ2 ... OBJn = COLL RECEVOIR ENT1;
 - ENT1 : expéditeur du message
 - OBJi : objets à recevoir (même ordre que le « COLL ENVOYER » correspondant)
 - Communication synchrone bloquante point à point
 - L'opérateur ne quitte pas tant que le message n'a pas été reçu

- **Communication non-bloquante**

- Utilisation des assistants

- ASSI ENT1 COLL ENVOYER ENT1 OBJ1 OBJ2 ... OBJn;

- OBJ1 OBJ2 ... OBJn = ASSI ENT1 COLL RECEVOIR ENT1;

- Communication vers soi-même impossible sans les assistants

- **Communication globale**

- Utilisation des tables esclaves et des boucles automatiques de ASSI TOUS

- Envoi vers tous les collaborateurs

- ASSI TOUS COLL ENVOYER RANGCOLL TABOBJ1 TABOBJ2 ... TABOBJn;

- Réception depuis tous les collaborateurs

- TABOBJ1 TABOBJ2 ... TABOBJn = ASSI TOUS COLL RECEVOIR RANGCOLL;

- Gain : pas de verrouillage de segments entre collaborateurs
- ... mais temps de communication non-négligeable

- **Limitation des communications côté opérateur**
 - Cast3m ne modifie pas les objets, il en crée de nouveaux
 - Historique des communications
 - Limitation des communications en détectant les objets déjà envoyés

- **Limitation des communications côté utilisateur**
 - Envoi uniquement des objets nécessaires à la méthode
 - Envoi de modèles et de champs par élément plutôt que des rigidités
 - Utilisation de partitions regroupant les sous-domaines plutôt que les partitions Arlequin

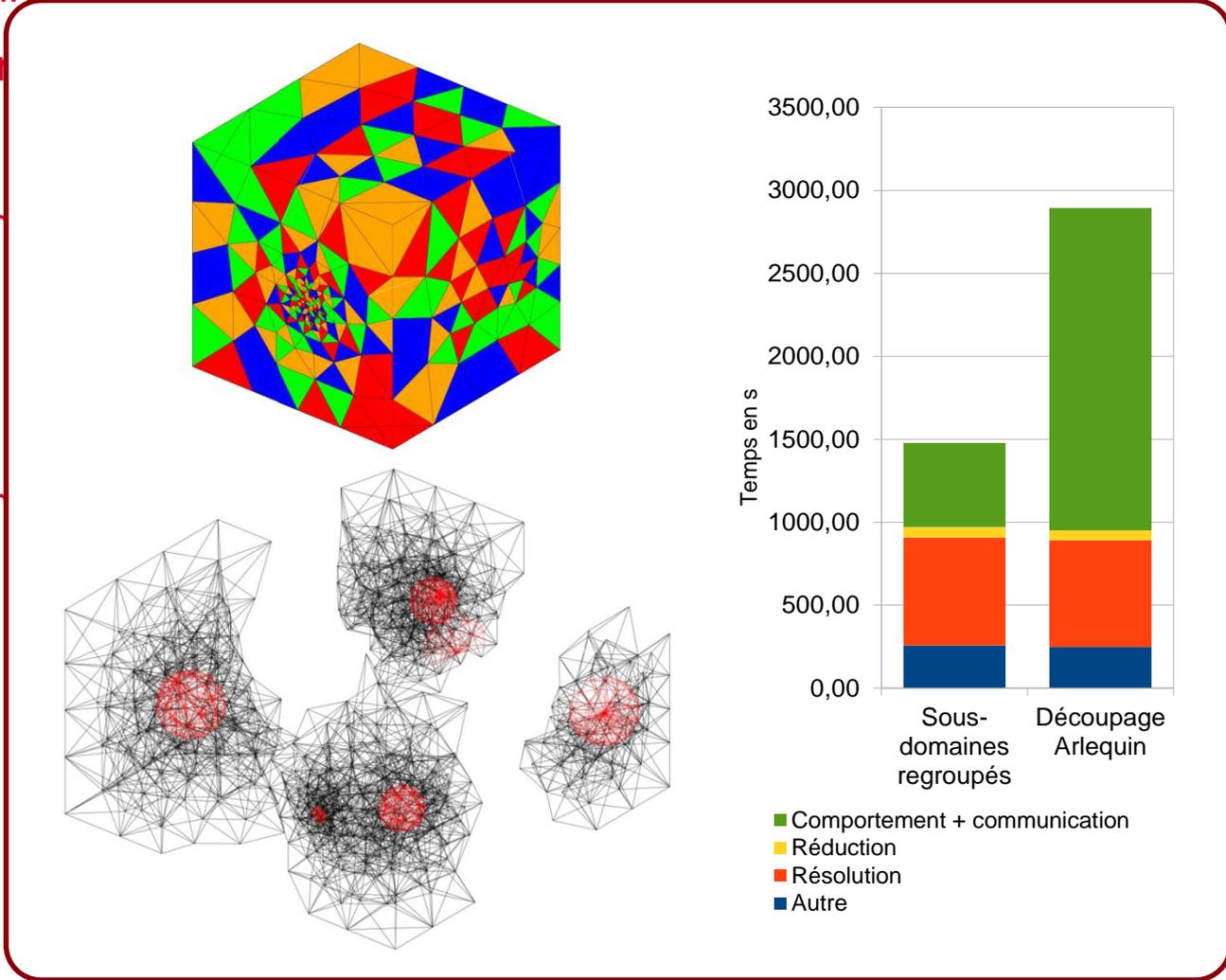
- Gain : pas de verrouillage de segments entre collaborateurs

• ...

• Lim



• Lim



- Gain : pas de verrouillage de segments entre collaborateurs
- ... mais temps de communication non-négligeable

- **Limitation des communications côté opérateur**
 - Cast3m ne modifie pas les objets, il en crée de nouveaux
 - Historique des communications
 - Limitation des communications en détectant les objets déjà envoyés

- **Limitation des communications côté utilisateur**
 - Envoi uniquement des objets nécessaires à la méthode
 - Envoi de modèles et de champs par élément plutôt que des rigidités
 - Utilisation de partitions regroupant les sous-domaines plutôt que les partitions Arlequin

- Utilisation transparente de l'opérateur
 - Objet reçu directement utilisable

- Intégrité référentielle
 - Echange d'objet Cast3m
 - Gestion de la suppression des objets
 - Invalidation des pointeurs dans l'historique lors de la suppression
 - Création des nœuds nécessaires
 - Envoi des dépendances des objets
 - Gestion des renumérotations de nœuds
 - Répercussion des changements dans les historiques
 - Prise en charge des configurations et mise à jour des coordonnées des nœuds

- Résolution d'un système linéaire
 - Méthode de Schur

- Principe
 - Condensation des sous-domaines sur l'interface entre les sous-domaines
 - Résolution du système condensé sur l'interface
 - Méthode simple mais pas forcément efficace
 - Equivalent à l'utilisation de super-éléments

- **Méthode de Schur**

- Résolution d'un système linéaire

- $KU = F \leftrightarrow \sum(K^i) U = \sum(F^i)$

- Décomposition des inconnues

- Inconnues interne : U_b

- Inconnues d'interface : U_Γ

- Condensation sur les inconnues d'interface

$$\begin{bmatrix} K_b^i & B^i \\ (B^i)^T & K_\Gamma^i \end{bmatrix} \begin{bmatrix} U_b^i \\ U_\Gamma \end{bmatrix} = \begin{bmatrix} F_b^i \\ \bar{F} \end{bmatrix}$$

$$U_b^i = (K_b^i)^{-1} (F_b^i - B^i U_\Gamma)$$

$$\sum (K_\Gamma^i - (B^i)^T (K_b^i)^{-1} B^i) U_\Gamma = \sum S^i U_\Gamma = \bar{F} - \sum (B^i)^T (K_b^i)^{-1} F_b^i = F_\Gamma$$

- Résolution du système condensé sur l'interface

$$K U = F \leftrightarrow \sum(S^i) U_\Gamma = \sum(F_\Gamma^i) \leftrightarrow S U_\Gamma = F_\Gamma$$

- Les calculs des compléments de Schur locaux S^i sont indépendants

- Algorithme

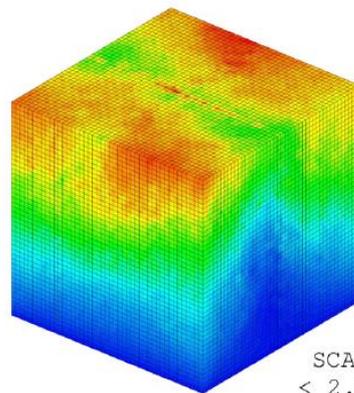
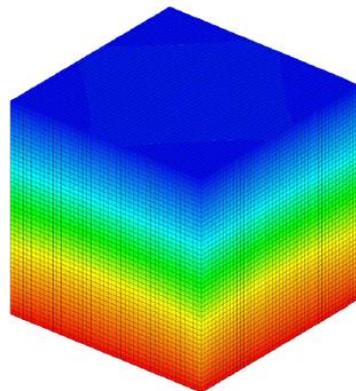
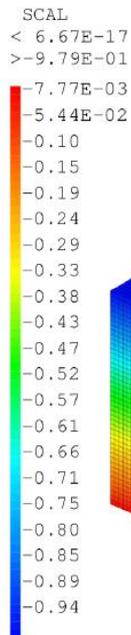
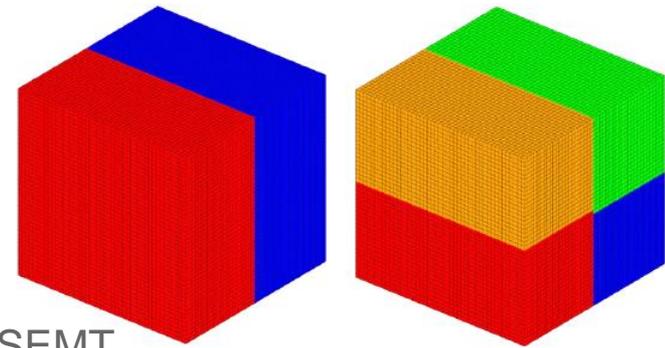
Construire les matrices de rigidités locales K_i	<code>mpart = part ndom mail; ripart = assi tous redu rig1 mpart;</code>
Envoyer les rigidités locales et les seconds membres aux collaborateurs	<u><code>assi tous coll envoyer rang mpart ripart fpart gart; mailo rilo flo gapart = coll recevoir 1;</code></u>
Calculer les compléments de Schur locaux S^i et les seconds membres condensés	<code>suplo = supe 'RIGIDITE' rilo galo; Slo = extr 'RIGIDITE' suplo; Fglo = supe 'CHARGE' suplo flo;</code>
Envoyer les matrices et seconds membres condensés	<u><code>coll envoyer 1 Fglo Slo; Spart Fgpart = assi tous coll recevoir rang;</code></u>
Assembler et résoudre le système d'interface $SU_\Gamma = F_\Gamma$	<code>S = et Spart; Fg = et Fgpart; Ug = reso S Fg;</code>
Envoyer les déplacements d'interface	<u><code>assi tous coll envoyer rang Ugpart fpart uglo flo = coll recevoir 1;</code></u>
Calculer les déplacements internes aux sous-domaines $U_i = (K_i)^{-1} (F_i - B_i U)$	<code>u = supe 'DEPLA' suplo uglo flo;</code>

- Regroupement des opérations dans deux procédures
 - Procédure pour le premier collaborateur REsolution PArallèle
 - DU = REPA NDOM RIGI FOR
 - Procédure des autres collaborateurs
 - REPA_ES;
- Utilisation simple
 - Même syntaxe qu'un RESO classique
 - DU = RESO RIGI FOR

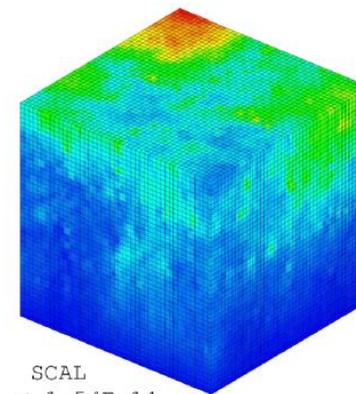
EXEMPLE D'UTILISATION : RESOLUTION D'UN SYSTEME LINEAIRE PAR LA METHODE DE SCHUR

- Application à un cube

- Cube unitaire
- $45 \times 45 \times 45 \approx 90\,000$ éléments
- Système bien conditionné
- Pas de difficulté particulière
- Résolution sur 2 ou 4 machines sur un cluster du SEMT
- Pas de différence de résultat avec RESO (différence à 10^{-14} près)

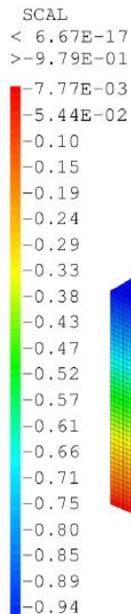


SCAL
< 2.55E-14
> 0.00E+00



SCAL
< 1.54E-14
> 0.00E+00

- Appl

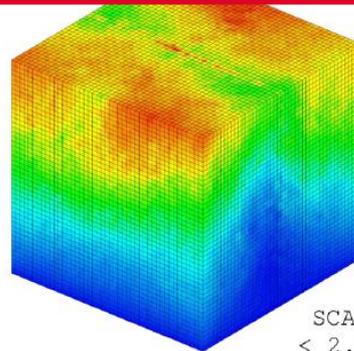
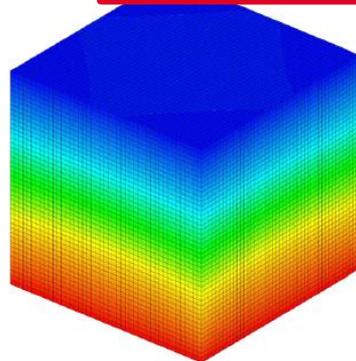
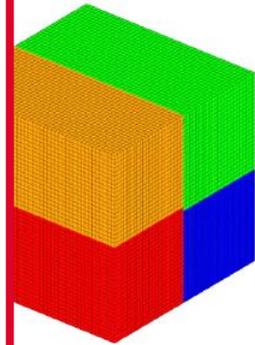


```

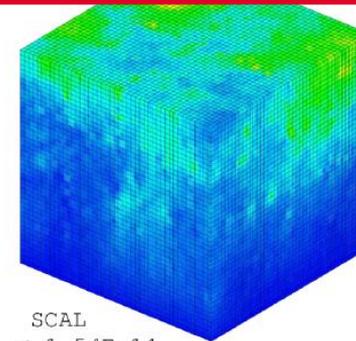
opti dime 3 elem cub8;
**maillage
a1=0. 0. 0.; a2=1. 0. 0.; a3 =0. 1. 0.;
**nombre d elements
nbel=45;
d1=a1 d nbel a2;
s1=d1 tran nbel (0. 1. 0.);
msh = s1 volu nbel tran (0. 0. 1.);
nua1 = msh poin plan (0. 0. 1.) (1. 0. 1.) (0. 1. 1.) 1.e-05;
s2 = (enve msh) elem appuy strict nua1;
**modele et materiau
mdl=mode msh mecanique elastique;
mtx=mate mdl 'YOUN' 1. 'NU' 0.3;
**chargement et cl
ft=press mass mdl 1. s2;
cl=(bloq UX UY UZ s1);
**rigidite
kt=(rigi mdl mtx) et cl;
**inversion de kt
du2 = reso kt ft;
**Tracé de la solution
trac (exco 'UZ' du2) msh;

fin;

```



SCAL
< 2.55E-14
> 0.00E+00

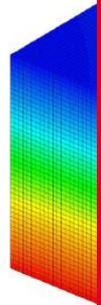


SCAL
< 1.54E-14
> 0.00E+00

- Appl

- C
- 4
- S
- F
- F
- F

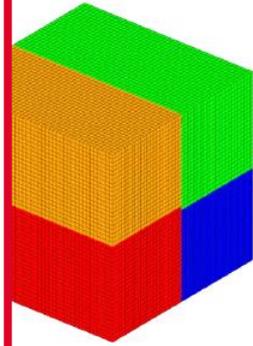
SCAL
 < 6.67E-17
 > -9.79E-01
 -7.77E-03
 -5.44E-02
 -0.10
 -0.15
 -0.19
 -0.24
 -0.29
 -0.33
 -0.38
 -0.43
 -0.47
 -0.52
 -0.57
 -0.61
 -0.66
 -0.71
 -0.75
 -0.80
 -0.85
 -0.89
 -0.94



```

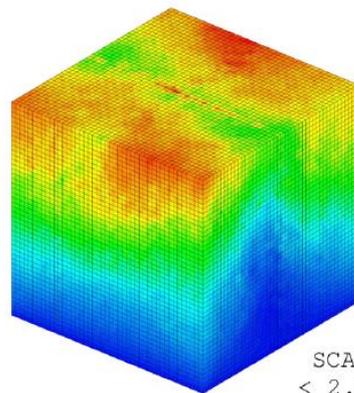
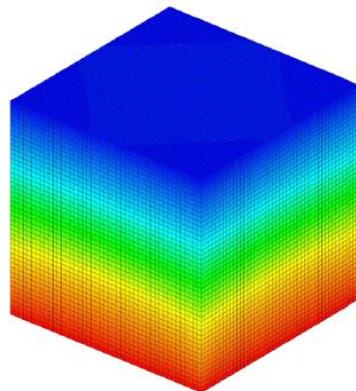
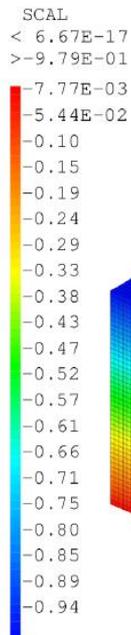
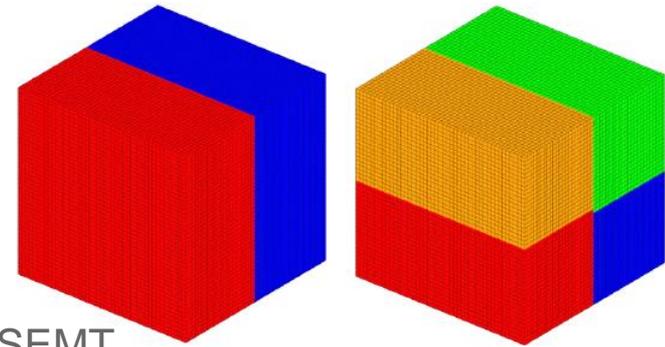
opti dime 3 elem cub8;
coll debut; totproce = coll nombre; nproce = coll rang;
si (nproce ega 1);
  **maillage
  a1=0. 0. 0.; a2=1. 0. 0.; a3 =0. 1. 0.;
  **nombre d elements
  nbel=45;
  d1=a1 d nbel a2;
  s1=d1 tran nbel (0. 1. 0.);
  msh = s1 volu nbel tran (0. 0. 1.);
  nua1 = msh poin plan (0. 0. 1.) (1. 0. 1.) (0. 1. 1.) 1.e-05;
  s2 = (enve msh) elem appuy strict nua1;
  **modele et materiau
  mdl=mode msh mecanique elastique;
  mtx=mate mdl 'YOUN' 1. 'NU' 0.3;
  **chargement et cl
  ft=press mass mdl 1. s2;
  cl=(bloq UX UY UZ s1);
  **rigidite
  kt=(rigi mdl mtx) et cl;
  **inversion de kt
  du = repa totproce kt ft;
  du2 = reso kt ft;
  **Tracé de la solution
  trac (exco 'UZ' du) msh;
  **Calcul de l'erreur
  u_err = du - du2;
  nuerr = ABS (EXCO UX u_err);
  nuerr = nuerr + (ABS (EXCO UY u_err));
  nuerr = nuerr + (ABS (EXCO UZ u_err));
  trac nuerr msh;
sinon;
  REPA_ES;
finsi;
coll fin;
fin;

```

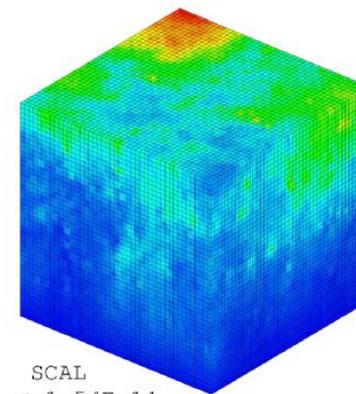


- Application à un cube

- Cube unitaire
- $45 \times 45 \times 45 \approx 90\,000$ éléments
- Système bien conditionné
- Pas de difficulté particulière
- Résolution sur 2 ou 4 machines sur un cluster du SEMT
- Pas de différence de résultat avec RESO (différence à 10^{14} près)



SCAL
< 2.55E-14
> 0.00E+00



SCAL
< 1.54E-14
> 0.00E+00

- **Nouvel opérateur COLL**
 - Permet des communications entre plusieurs instances de Cast3m
 - Utilisation bloquante ou non-bloquante en utilisant les assistants
 - Utilisation des tables parallèles et de l'instruction « ASSI TOUS » pour réception ou envoi global
 - Disponible dans la version du jour et 2015
- **Première méthode**
 - Méthode de Schur
 - Méthode applicable en calcul distribué avec l'opérateur COLL
 - Encapsulable dans une procédure
- **Intérêt des collaborateurs**
 - Boite à outil pour la mise en place de méthode numérique parallèle
 - Plusieurs serveurs Esope en parallèle
- **Travaux à venir**
 - Intégration à PASAPAS
 - Autres méthodes de résolution et de couplage
 - Détection des systèmes déjà résolu
 - Support d'objets supplémentaires

Merci de votre attention

