

Contexte et Problématique

- Applications scientifiques denses
- Temps de calcul importants, limitation mémoire
- Parallélisation semi-automatique
- Traiter des problèmes de très grande taille
- Réduire le temps d'exécution global

Dimensions

- Distribution des calculs
- Distribution des données
- Communications
- Facilité de programmation

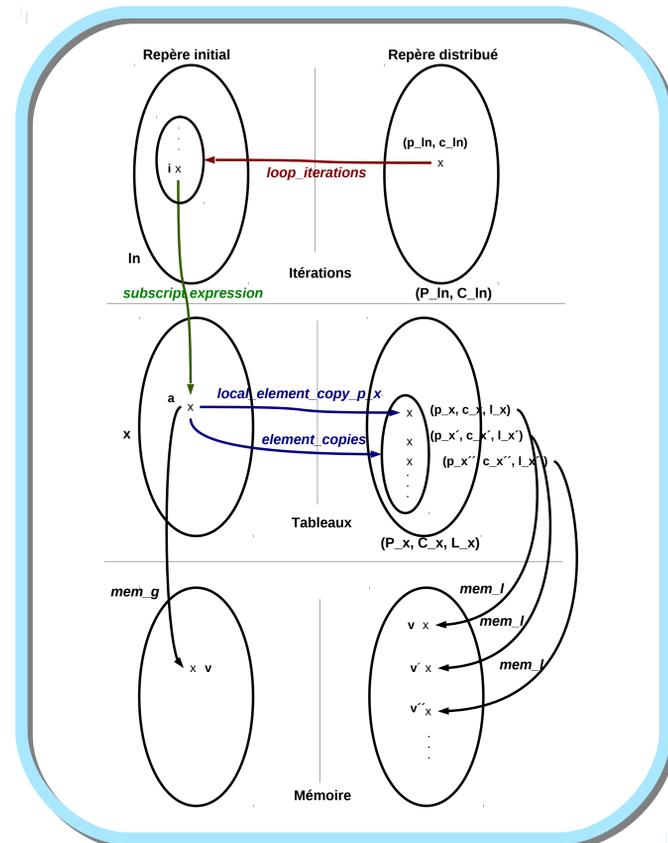
Transformation

- Source à source
- Langage C
- Code en entrée : code séquentiel + OpenMP + distribut
- Code généré : OpenMP + MPI (runtime STEP)

Travaux liés

Distribution et changement de repère

Langage à base de directives	Solution proposée	Architecture	Distribution des calculs	Distribution des données	Communications
HPF	Description de la distribution des données Dérivation de la répartition des calculs et des communications à partir de la distribution des données	Mémoire distribuée	-	++	+
XcalableMP	Description de la distribution des données et des communications Dérivation de la répartition des calculs à partir de la distribution des données	Mémoire distribuée	-	++	-
OpenMP	Description de la répartition des calculs à exécuter en parallèle	Mémoire partagée	++	-	-
HMPP	Description des noyaux de calcul à exporter et à exécuter sur GPU	GPU	+	-	+



Objectif: proposer un modèle de distribution des calculs et des données et générer de façon semi-automatique un code parallèle efficace en consommation mémoire et en temps d'exécution

Notre modèle de distribution

1. Distribution des calculs sans halo
 $i = Bp_c + Bp + l + L.1$
#pragma step gridify
#pragma omp parallel for
2. Distribution des données avec halo
 $a = Bp_c + Bp - H_low + l$
#pragma step distribute

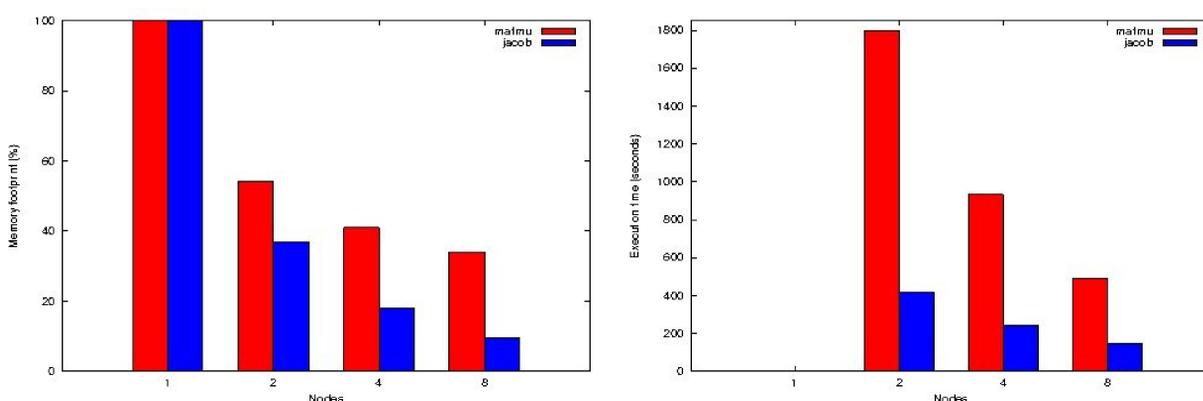
Implémentation

- Prototype implémenté dans STEP (basé sur PIPS)
- Résumés des communications en utilisant les régions de tableaux de PIPS
- Communications générées :
 - non bloquantes
 - en cas de définitions utilisées dans le futur du calcul
 - limitées aux voisins

Nos Contributions

- Distribution des calculs pour des nids de boucles parallèles à profondeur quelconque
- Distribution des données avec extension de l'espace de stockage : halos
- Génération de code source parallèle proche du code initial avec traduction des accès globaux en accès locaux
- Génération des communications pour les mises à jour des halos

Résultats préliminaires



Conclusion et suite

- Proposition d'un modèle de distribution des calculs et des données
- Propriétés de correction du code généré
- Implémentation dans STEP (PIPS)
- Génération de code parallèle hybride
- Premières expérimentations concluantes
- Prouver l'intérêt de la solution sur davantage d'applications
- Modèle de coût pour la distribution