

# Sujet : Gestion et optimisation de la structuration de données multidimensionnelles dans des stencils parallèles et distribués

**Responsables :** Denis Barthou, Olivier Aumage

**Téléphones :** 05 24 57 41 16, 05 24 57 41 19

**Courriels :** [denis.barthou@labri.fr](mailto:denis.barthou@labri.fr), [olivier.aumage@inria.fr](mailto:olivier.aumage@inria.fr)

**Thème:** Support exécutifs hautes performances

**Équipe:** Satanas/Runtime

**Financement:** Projet ANR JST (franco-japonais)

**Mot-clés :** parallélisme, calcul distribué, stencil, compilation

## Présentation du sujet :

### Contexte

L'équipe Runtime (INRIA/LABRI) travaille à la conception de supports d'exécution et d'outils d'analyse d'applications scientifiques sur des machines de calcul parallèles et distribuées, dans le domaine que l'on nomme *calcul intensif*. Le but fondamental de ce logiciel est d'*assurer* et de *vérifier* la *portabilité des performances* des applications en permettant à celles-ci de *maximiser* l'exploitation du potentiel de performances des machines tout en *minimisant* ou en éliminant l'effort d'adaptation nécessaire à l'utilisation de telles applications sur une machine donnée.

### Problématique

De nombreux problèmes de simulation numérique de phénomènes physiques peuvent se ramener à un schéma général de calcul appelé en Anglais *stencil* (pochoir en Français). La version n+1 d'un ensemble de points est calculée simultanément en tout point P par une fonction locale de la valeur de ce point et de ses voisins immédiats, selon un motif répété à l'identique pour chaque point (d'où l'appellation de pochoir). Pour des algorithmes simples travaillant sur des données de faible dimension, la programmation d'un stencil est relativement triviale. Lorsque les algorithmes se complexifient, lorsque le nombre de dimensions dépasse 2 ou 3, et surtout lorsqu'il devient nécessaire de paralléliser et/ou de distribuer les calculs, la programmation de tels stencils s'avère rapidement très difficile et sources d'erreurs.

Cependant, le caractère fortement répétitif et régulier des stencils offre de fait la possibilité d'une automatisation partielle du processus de développement. Plusieurs outils ont donc été développés pour permettre aux programmeurs d'exprimer des stencils avec un plus haut niveau d'abstraction, plus proche de la syntaxe mathématique du problème physique simulé, et plus éloigné des considérations techniques de la programmation impérative. De tels outils se chargent ensuite de générer de façon automatique le code source effectuant le calcul du stencil proprement-dit. Le langage QIRAL développé au sein de l'équipe Runtime est l'un de ces outils.

Afin d'exploiter pleinement les plateformes de calculs modernes, il est nécessaire de répartir la charge de calcul d'une part entre les nœuds physiques d'une plateforme et d'autre part --- à une échelle plus fine --- entre les cœurs de calcul d'un nœud (et à plus long terme, entre les cœurs généralistes et les unités accélérées (GPUs) d'un nœud). Or, la structuration des données (le choix du *data layout* en Anglais) joue un rôle clé dans l'efficacité des calculs de stencils parallèles et distribués.

### Travail demandé

- Dans un premier temps, il s'agit d'effectuer un travail bibliographique sur le sujet. De récents articles [1,2,3] abordent le problème de la génération de code pour des calculs de stencils en partant d'un langage de plus haut niveau.
- Dans un deuxième temps, l'objectif est de concevoir un cadre permettant de définir automatiquement à partir d'une représentation de stencil une répartition hiérarchique du calcul suivant les niveaux de parallélisme. Cela comprendra notamment la génération d'un *data layout* adapté au nombre de dimensions et conjointement d'une infrastructure de communication permettant la mise à jour efficace des voisinages aux frontières des blocs. Une première idée est de s'appuyer sur le toolkit Global Arrays, une bibliothèque permettant la manipulation de tableaux multi-dimensionnels distribués, pour l'expression du parallélisme entre nœuds et sur le langage OpenMP pour le parallélisme intra-nœud.

### Liens :

- [1] Physis: An Implicitly Parallel Programming Model for Stencil Computations on Large-Scale GPU-Accelerated Supercomputers. N. Matsuoka et al. SuperComputing 2011.
- [2] PATUS for Convenient High-Performance Stencils: Evaluation in Earthquake Simulations. OL Schenk et al. SuperComputing 2012.

- [3] The Pochoir Stencil Compiler. Leiserson et al. SPAA conference 2011.
- Équipe Runtime : <http://runtime.bordeaux.inria.fr/>
- Global Arrays : <http://www.emsl.pnl.gov/docs/global/>