

# Passage vers le modèle de programmation par tâches hétérogènes pour l'exploitation optimale des supercalculateurs actuels et futurs : Analyse et Méthodologie

Manuel Ferat <sup>1,2</sup>

<sup>1</sup> CEA, DAM, DIF,  
F-91297 Arpajon, France  
manuel.ferat@cea.fr

<sup>2</sup> LICIIS - Laboratoire d'Informatique en Calcul Intensif et Image pour la Simulation  
Université de Reims Champagne Ardenne, France

---

## Résumé

Les supercalculateurs des générations actuelles et futures reposent sur des architectures massivement parallèles et hétérogènes. Pour exploiter ces architectures au mieux, les développeurs doivent disposer des modèles de programmation parallèle adaptés pour exposer suffisamment de parallélisme, exploiter au mieux des mécanismes asynchrones et de l'hétérogénéité des ressources de calculs. Le modèle de programmation par tâches a été introduit depuis plusieurs années comme une solution prometteuse, mais il reste sous-utilisé pour le développement de nouvelles applications. Dans ces travaux, nous proposons d'analyser les différents avantages et les freins du passage vers le modèle de programmation par tâches. De ce fait, nous introduisons une méthodologie en trois étapes permettant d'effectuer le changement de sémantique pour porter une application MPI+OpenMP vers un modèle de programmation par tâches. (i) La première étape de la méthode détaille comment découper une application en blocs pour en définir un ensemble de tâches et leurs dépendances associées. (ii) Ensuite, l'exécution de tâches sur machines hétérogènes est introduite à l'aide des constructions `target` OpenMP. Cette étape permet l'identification des portions à déporter sur accélérateur de calcul et la spécification des mouvements de données associés. (iii) La dernière étape est l'étude du paramétrage fin de l'application au travers d'une étude de grain, pour déterminer la taille optimale des tâches, et d'une étude de la répartition des processus MPI, pour exploiter au mieux l'architecture visée. Enfin, nous illustrons la mise en œuvre de cette méthode sur la mini-application miniFE et exposons les gains de performances obtenus au travers d'un weak-scaling, avec une réduction de temps d'exécution allant jusqu'à 2,5 fois.

**Mots-clés :** Portage, Tâches, OpenMP, Hétérogénéité

---