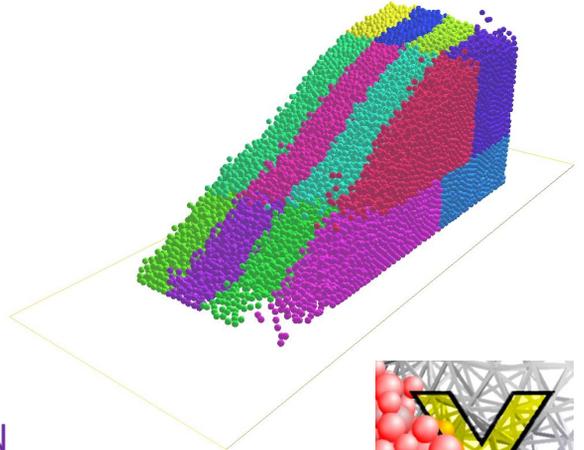
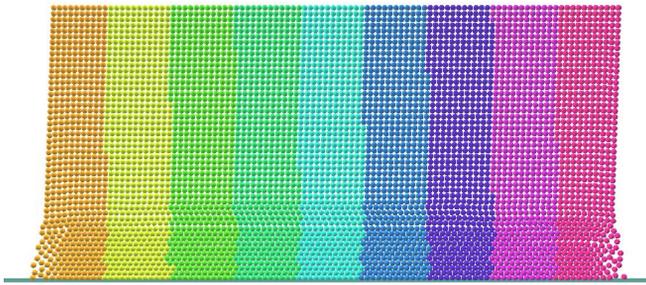


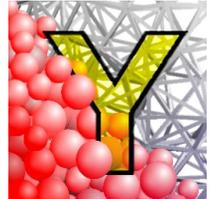


Hackathon du HPC



Yade-MPI OPTIMISATION

Calcul distribué MPI pour la dynamique de systèmes particulaires



Contexte

La simulation mécanique de systèmes formés de nombreuses particules en interaction repose sur la "méthode des éléments discrets". La modélisation de ces systèmes hétérogènes nécessite de détecter et simuler les interactions entre particules, au prix d'une complexité élevée (jusqu'à N^2 , où N est le nombre de particules).

Outils employés

- Yade - moteur dynamique open source
- mpi4py - wrapper python pour les communications MPI
- perf_event - outil de profilage linux
- NIX - distribution rapide du code
- OAR - soumission de travaux par lots

Objectifs

1. Examiner la scalabilité forte et faible pour un nombre de particules N_p entre $10e4$ et $10e6$ et un nombre de coeurs N_c compris entre 1 et 100.
2. Produire un bilan détaillé des coûts associés aux différentes communications.
3. Améliorer l'implémentation initiale afin de réduire le temps d'exécution du script de test.
4. Quantifier l'optimisation par la borne supérieure C_{max} du nombre de Cundall ($N_p / \text{wallTime}$) sur l'ensemble des combinaisons N_p/N_c .

Resultats

- Accélération des performances maximales de 300%
- Diminution du temps de passage des messages de 88%
- Validité des performances constantes pour une large gamme de tailles de domaines et de décompositions
- Augmentation du nombre de Cundall max de 2.1

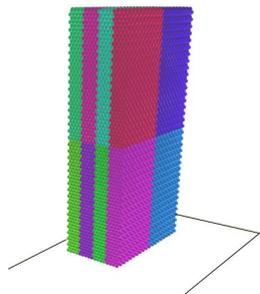
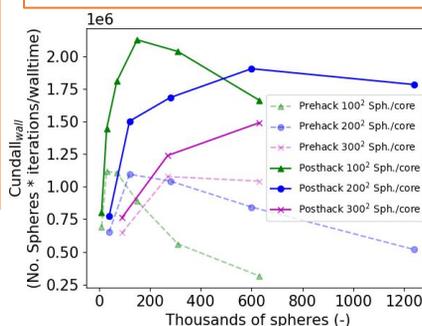


Figure 1. Comparaison des performances avant vs après le hackathon (left) ; exemple de décomposition du domaine (right).

Méthodes d'optimisation :

Améliorer la fusion des données

L'envoi des données des particules des N sous-domaines au processus maître a été amélioré par l'utilisation de MPI Gather_v (précédemment réalisé par des envois en série).

Optimiser l'envoi de la scène du maître aux noeuds de calcul

La distribution des informations des particules aux noeuds de calcul a été accélérée en optimisant la quantité et le type des informations envoyées (précédemment des objets python sérialisés étaient envoyés). Cela a nécessité l'implémentation de getters/setters au niveau du coeur C++ du code Yade-DEM.

Outils de décomposition du domaine

Plusieurs outils de décomposition du domaine ont été développés pour Yade-MPI, notamment des méthodes d'optimisation manuelle, bisection (basée sur un arbre), nuage de points et grille des moindres carrés.

Yade-MPI

Robert Caulk, Deepak Kunhappan, Jean Pourroy, Thomas Guntz, William Cheveremont

CIMENT GRICAD Grenoble

Admin: Pierre-Antoine Bouttier
Yade consultant: Bruno Chareyre MCF UGA

19828.9 heures de utilisées

