Développement, optimisation, parallélisation de code

Journée calcul INRA IRSTEA

Sylvain Jasson MIA TOULOUSE

14 juin 2019

Le calcul scientifique : une définition

Discipline qui regroupe un ensemble de champs mathématiques et informatiques permettant la simulation numérique des phénomènes de la physique, chimie, biologie, et sciences appliquées. L'approche dun problème par le biais du calcul scientifique est une démarche globale, qui se passe en plusieurs étapes :

- Modélisation du système,
- Analyse théorique du modèle et étude de ses propriétés,
- Choix d'une méthode numérique adaptée aux propriétés théoriques du modèle,
- Implémentation informatique,
- Utilisation de la nouvelle méthode sur des cas réels.

Le calcul scientifique est donc par essence un domaine interdisciplinaire.

Le contenu de l'implémentation informatique

- ...
- Moyens de rendre les résultats fiables/partageables/reproductibles
- Moyens de visualisation des résultats
- Sauvegarde des résultats
- Environnement spécifique dont données et logiciels maison
- Gestionnaire de workflow
- Bases de données & Bibliothèques de code
- Communs (versionnés)
- Réseau (Accès aux données, au capteur, Échange entre les processus)
- OS
- Silicium (Flops, Opérations logiques, Mémoire)

Rappel : Optimiser c'est bien, avoir un résultat juste c'est mieux !

L'offre de calcul

Taille des données	Type de ressource	Nombre de CPU	Accès
10^3	Machine perso	2^3	Immédiat
10^3	Tier 3 (cluster de laboratoire)	$2^4 - 2^8$	Selon la politique locale
10^4	Tier 2 (mésocentre)	$> 2^{10}$	"Fair share" ou appel d'offre
10^5	Tier 1 (national)	$> 2^{12}$	Appel doffre
10^8	Tier 0 (Europe)	$> 2^{15}$	Appel doffre

Pour plus de détails se référer à cet article du cahier des techniques de l'INRA.

L'important est que chaque passage d'un tiers au suivant me donne des gains de temps réels.

Profiler son code

Le calcul le plus rapide de tous c'est celui qu'on ne fait pas. Mais cela peut demander d'investir un peu de temps au départ.

Parfois se poser la bonne question mathématique permet de gagner beaucoup de temps de calcul.

Faire l'effort d'utiliser un langage compilé peut aussi être un investissement rentable.

Il existe des outils de profilage depuis fort longtemps. Ils permettent de savoir quels sont les parties du code qui consomment. Et donc quels algorithmes nécessitent des efforts, à la condition de les faire tourner de manière réaliste.

Exemple : Choix d'un algorithme de tri

On voudrait un tri qui :

- Soit stable : les valeurs équivalentes ne sont pas déplacées
- Soit économe en mémoire : travaille sur place
- Limite les comparaisons (au pire $O(n \log n)$)
- Limite les mouvements (au pire $O(n)$ échanges de valeurs)
- Exploite le cas où l'ensemble est presque trié
- Exploite le cas où l'ensemble contient peu de valeurs distinctes

Et on aurait une attitude incatatoire...

Exemple : Choix d'un algorithme de tri

Nom	Bogo Sort	Bubble Sort	Quick Sort	Spagetti Sort
Efficacité	$O(n!)$	$O(n^2)$	$O(n \log n)$	$O(n)$
Pire cas	Non borné	n^2	n^2	n

Je vous invite à aller voir une page très pédagogique ici

Dualité mémoire - CPU

Quand j'ai des calculs qui reviennent, je peux avoir intérêt à conserver le résultat d'un fois sur l'autre (surtout quand c'est facile à faire, memoization à coup de décorateur par exemple).

Ce n'est pas une opération magique :

- Est-elle justifiée ? (Étudions l'algorithme)
- Quel volume vais-je garder ? (Interrogeons les données)
- Dans quelle mémoire le mettre ? (Demandons à l'admin sytème)

J'ai mis dualité dans le titre du transparent, mais il faut que je rajoute pas les questions de bande passante si jamais je voulais partager mes résultats entre calculateurs

Dualité distribution - communication

Quand j'ai des calculs qui reviennent, qui se parallélisent bien (i.e. j'ai des étapes relativement indépendantes les unes des autres) je peux avoir intérêt à les dispatcher puis à collecter le résultat

Ce n'est pas non plus une opération magique :

- Est-elle justifiée ? (Étudions les dépendances)
- A-t-elle des effets de bord ? (Étudions l'algorithme, en particulier quand des simulations sont en jeu)
- A quelle distance vais-je transférer (du μm au km) ? (Demandons à l'admin système)
- Sachant la distance, et le débit, que vais-je transférer ? (Retour à la question précédente)

Pour Résumer

Le développement de codes scientifiques efficace nécessite d'avoir une approche globale et largement transdisciplinaire de la question à traiter.

Il est rare qu'une solution domine toutes les autres existe, il faut faire des compromis au mieux du matériel disponible.

Et ces compromis sont rarement définitifs...

Place à la discussion