Parallel Computing for Data Science
Lab x004 : Équilibrage de charge

Jairo Cugliari

S1 2020–2021

1 Répartition des tâches

Lire le code qui corresponde à l'exercice 1 dans le fichier adjoint.

1. Que fait la fonction `dormir` ?
2. Comment sont distribuées les tâches entre les unités de calcul?
3. Peut-on réduire le temps total de calcul avec une répartition alternative des tâches ?

2 Ordonnancement statique des tâches avec durée connue

Nous allons utiliser le code vu en cours pour évaluer la perte de performance d'un calcul en parallèle avec un équilibrage de charge mal reparti. Le but est double : d'une part nous cherchons à minimiser le temps total du calcul réalisé, d'autre part nous souhaitons réduire le temps qu'une unités de calcul est inactive.

Utiliser le code fourni en cours (exemple Mutual Outlinks) pour mesurer le temps de calcul sur les données simulés selon est indiqué dans le code.

Les stratégies à évaluer sont:

1. Calcul séquentiel.
2. Calcul en parallèle avec répartition de la charge
   (a) par lignes de la matrice: de la plus courte à la plus longue.
   (b) par lignes de la matrice: de la plus longue à la plus courte.
   (c) par bloques à l'aide de la fonction `clusterSplit`.
   (d) par bloques de manière aléatoire.

   Quels sont vos conclusions?

3 iris dataset is back !

Nous allons compléter les expériences numériques du Lab x003 concernant la calibration d'une méthode d'analyse de données par la méthode d'échantillonnage du type "leave-one-out".

Nous nous plaçons dans un contexte où la calibration doit être faite sur plusieurs jeux de données. Le code adjoint fourni la fonction `simuData` qui permet de générer des données de taille arbitraire à partir des données `iris`.

Deux situations sont à considerer

- Cas idéal : jeux de données de même taille (disons 200 points) et nombre de jeux de données multiple du nombre d'unités de calcul (disons 8)
- Cas réel : jeux de données de tailles 1000, 200, 1000, 200, 1000, 200, 250, 100, 100, 100

Mesurez le temps de calcul de chacune des stratégies de calcul parallèle ci-dessous:

1. par ordonnancement statique.
2. par ordonnancement dynamique à gros-grain.
3. par ordonnancement dynamique à gros-grain.
4. par aplatissement de l'imbrication avec `foreach`.

Pour les différents approches, chronométrez le temps d'exécutions et comparez.