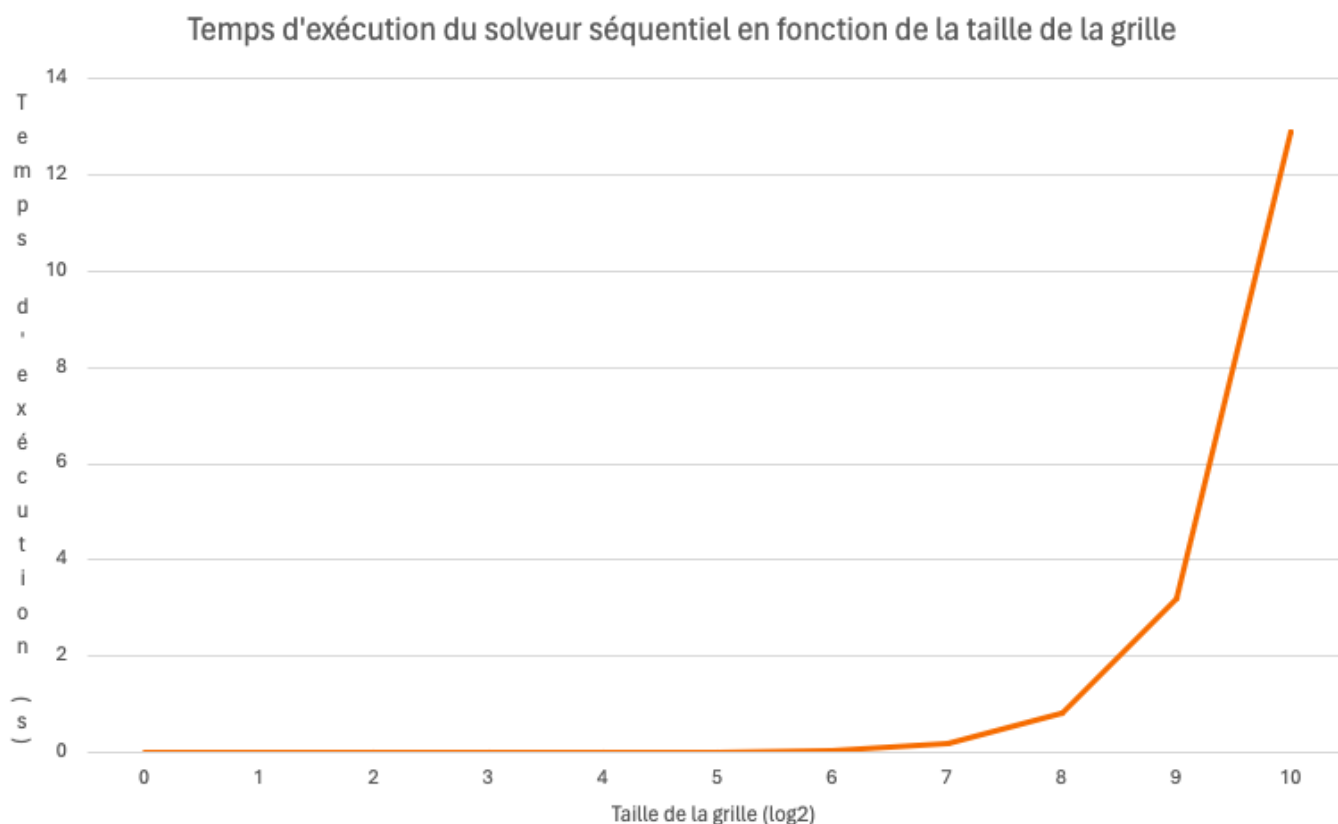


Rapport BE MPI ACCOU Martin LALONNIER Lucas

Q 1.2



Le temps de calcul du solveur séquentiel explose donc très rapidement, puisqu'on dépasse la dizaine de secondes pour une taille de grille égale à 1024. On comprend donc la nécessité de paralléliser le solveur, étant donné que les tailles de grilles usuelles ont des ordres de grandeur bien plus élevés que le millier.

Q 1.3

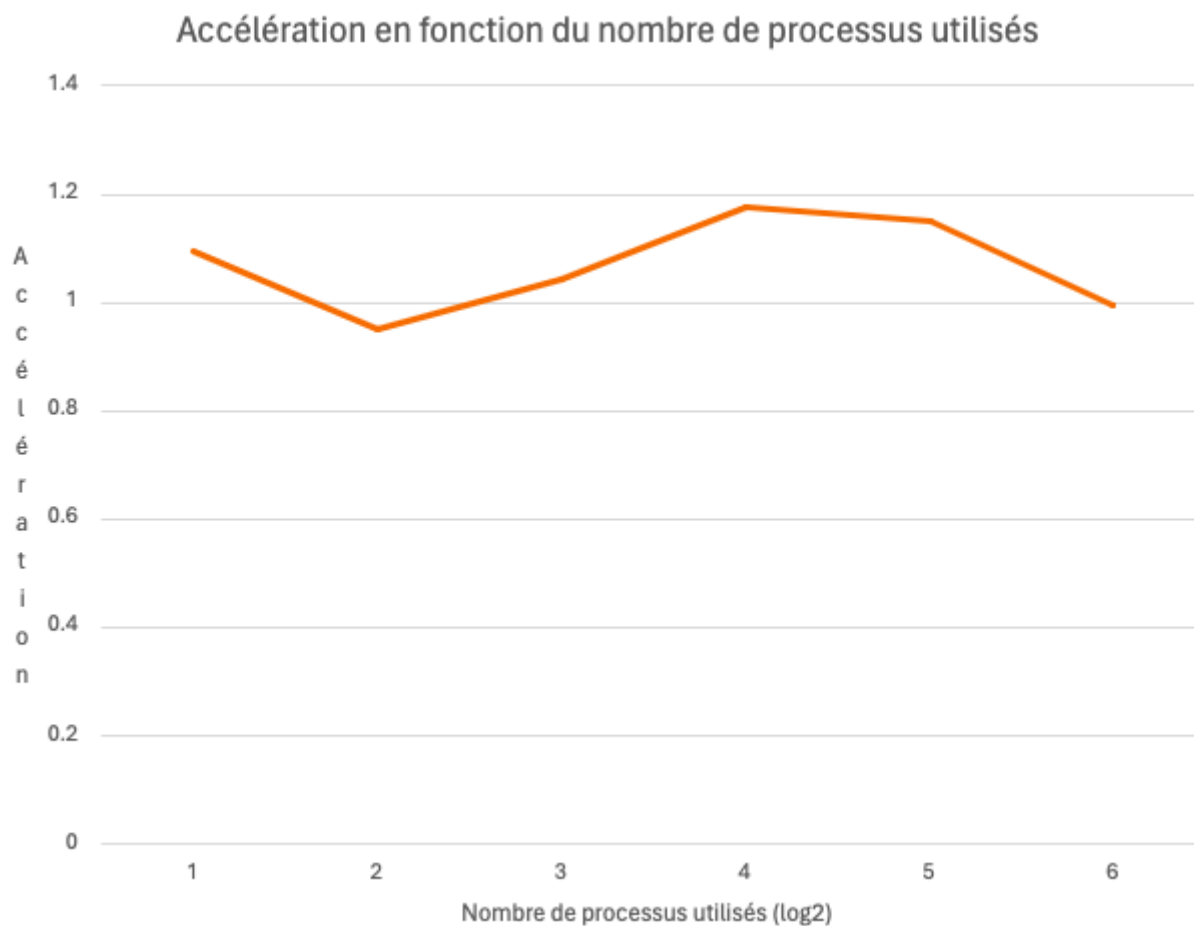
Nous avons d'abord passé beaucoup de temps avant de comprendre la structure du code, et notamment le fait qu'on n'a pas à subdiviser nous même la grille à la main. Ensuite, nous avons d'abord eu l'idée d'utiliser un GATHER pour rassembler les lignes dans le processus de rang 0, mais cela n'a pas marché car aucune version de GATHER ne permet de gérer le chevauchement des blocs à collecter.

Nous avons finalement géré les communications à la main avec Isend et Recv.

On doit être particulièrement vigilant au premier et au dernier bloc, dont la gestion des lignes adjacentes est différente d'un bloc interne à la grille.

Q 1.4

Le nombre de domaines selon X est fixé à 1 ici.



On n'observe pas de gains notables fournis par la parallélisation.

Q 1.5

On remarque que la taille de la grille à partir de laquelle la parallélisation vaut la peine est 1024 lorsqu'on travaille avec 2 processus. La taille limite est aussi 1024 avec 4 processus.

Q 2.1

On utilise REDUCE ici pour calculer la température moyenne.

Ceci est licite car la moyenne des moyennes particulières est bien la moyenne globale.

Q 3.2

On a utilisé une méthode qui n'est pas basée sur des types dérivés. Notre méthode marche mais n'est clairement pas efficace.