OS202 - Programming Parallel Computers

TP2

2 février 2026

NOUKOUA TATMFO Maëva Sandy

2025–2026

Table des matières

1 Parallélisation ensemble de Mandelbrot

1.1 Méthodologie de mesure

On mesure le temps total $T(p)$ pour p processus MPI (même taille d'image et même $N_{\max}$), puis on calcule :

$$S(p) = \frac{T(1)}{T(p)}.$$

1.2 (1) Répartition statique par blocs (lignes contiguës)

Chaque processus r calcule un bloc de lignes contiguës :

$$j_{start} = \left\lfloor \frac{rH}{p} \right\rfloor, \qquad j_{end} = \left\lfloor \frac{(r+1)H}{p} \right\rfloor.$$

Les résultats sont rassemblés sur le rang 0 (ex. `MPI_Gatherv`) pour reconstruire et sauvegarder l'image.

Interprétation : speedup limité par (i) surcoûts MPI et (ii) déséquilibre de charge car certaines lignes coûtent plus cher (frontière de l'ensemble).

1.3 (2) Meilleure répartition statique (ex. cyclique)

Répartition cyclique : le rang r calcule les lignes j telles que $j \equiv r \pmod p$. Cela répartit mieux les lignes coûteuses entre processus.

Comparaison : on compare $T_{bloc}(p)$ et $T_{cycl}(p)$ puis les speedups. **Limite :** stratégie dépendante de l'image/zoom ; rassemblement plus complexe (lignes non contiguës).

1.4 (3) Stratégie maître–esclave (répartition dynamique)

Le rang 0 distribue dynamiquement des lignes (ou paquets de lignes) aux autres rangs. Chaque esclave calcule et renvoie le résultat, puis redemande du travail. Un message “STOP” termine les esclaves.

Interprétation : meilleur équilibrage quand la charge est irrégulière, mais surcoût de communication ; le maître peut devenir un goulot si la granularité est trop fine.

1.5 Conclusion

La répartition par blocs est simple mais sensible au déséquilibre. La cyclique améliore l'équilibrage statique au prix d'un rassemblement plus complexe. Le maître–esclave fournit en général le meilleur équilibrage pour Mandelbrot, mais nécessite un compromis sur la taille des tâches pour limiter le coût des communications. rocessus) + MPI (inter-processus).

2 Entraînement pour l'examen écrit

On considère qu'une fraction $P = 0{,}9$ du temps d'exécution d'un programme est parallélisable, tandis que la partie séquentielle représente $1 - P = 0{,}1$ du temps total.

2.1 Accélération maximale selon la loi d'Amdahl

La loi d'Amdahl donne l'accélération maximale théorique pour un nombre de nœuds $n \gg 1$:

$$S_{\max} = \frac{1}{1-P}.$$

En remplaçant $P = 0{,}9$, on obtient :

$$S_{\max} = \frac{1}{0{,}1} = 10.$$

L'accélération maximale théorique est donc égale à 10, indépendamment du nombre de nœuds utilisés.

2.2 Choix d'un nombre raisonnable de nœuds

Au-delà d'un certain nombre de nœuds, l'augmentation des ressources de calcul n'apporte plus de gain significatif en raison de la partie séquentielle incompressible et des surcoûts de communication. Il est donc raisonnable de choisir un nombre de nœuds proche de la zone où l'accélération commence à saturer, typiquement de l'ordre de 8 à 16 nœuds, afin d'éviter un gaspillage de ressources CPU.

2.3 Accélération observée en pratique

Alice observe expérimentalement une accélération maximale de 4 en augmentant le nombre de nœuds. Cette valeur, inférieure à la limite théorique donnée par la loi d'Amdahl, s'explique par les surcoûts liés aux communications, à la synchronisation et au déséquilibre de charge, qui ne sont pas pris en compte dans le modèle idéal.

2.4 Accélération selon la loi de Gustafson

On suppose maintenant que la quantité de données à traiter est doublée et que la complexité de l'algorithme parallèle est linéaire. La loi de Gustafson s'écrit :

$$S_G(n) = n - (1 - P)(n - 1).$$

Pour $n = 4$ et $P = 0{,}9$, on obtient :

$$S_G(4) = 4 - 0{,}1 \times 3 = 3{,}7.$$

Alice peut donc espérer une accélération maximale d'environ 3,7, ce qui est cohérent avec l'accélération observée en pratique.