

# Laboratoire de High Performance Coding

## semestre printemps 2025

### Laboratoire 7 : Parallélisme des tâches

Temps à disposition : 4 périodes (2 séances de laboratoire).

## 1 Introduction

Ce laboratoire a pour objectif de vous familiariser avec les techniques et outils de parallélisation des tâches. Vous apprendrez à implémenter des solutions multithreadées pour des problèmes de traitement de données.

## 2 Calcul multithreadé de la fréquence de k-mers dans un fichier texte

Dans cette activité, vous devez analyser un fichier texte (par exemple contenant les chiffres de Pi, de l'ADN, ou tout autre texte brut volumineux) afin de calculer la fréquence de toutes les sous-séquences ("k-mers") de taille  $k$ , où  $k$  est fourni en paramètre. Par exemple, pour  $k = 3$  sur la chaîne ABCDABC, les k-mers extraits sont : 2xABC, BCD, CDA, DAB.

### 2.1 Objectifs

- Améliorer la logique et le traitement de la version mono-threadée fournie.
- Implémenter une version multithreadée (OpenMP ou pthreads).
- Comparer les performances entre votre version multithreadée et la version mono-threadée améliorée.

Le code qui vous est fourni lit naïvement le fichier donné et compte le nombre d'occurrences de chaque k-mer de longueur  $k$ . Vous êtes libre de la modifier, d'ajouter ou de supprimer ce que vous voulez. Veuillez simplement à ne pas changer le comportement de l'usage du programme (son interface en ligne de commande).

Vous devrez également modifier le `CMakeLists.txt` en conséquence.

Voici quelques fichiers volumineux contenant des chiffres de Pi que vous pouvez utiliser pour tester votre solution :

- <https://pi2e.ch/blog/2017/03/10/pi-digits-download/#download>
- <https://calculat.io/en/number/download-pi-digits-files-txt-zip>

Mais vous êtes libre d'utiliser n'importe quel fichier texte de votre choix comme entrée.

### 3 DTMF!

---

Pour conclure ce laboratoire, nous vous invitons à revenir une dernière fois sur votre code. Cette étape finale vise à en extraire tout le potentiel en appliquant une parallélisation efficace de votre traitement.

### 4 Travail à rendre

---

- Deux programmes fonctionnels et compilables :
  - Un pour l'activité sur les k-mers.
  - Un pour l'activité DTMF.
- Un rapport structuré en deux parties :
  - **Première partie — Analyse des k-mers :**
    - \* Une explication des éléments inefficaces dans le code fourni, et des améliorations que vous y avez apportées.
    - \* Une analyse des performances de votre version mono-threadée.
    - \* Une description de la stratégie de parallélisation mise en œuvre : répartition du travail entre threads, traitement des cas limites, zones de chevauchement, etc.
    - \* Une comparaison détaillée entre les performances des versions mono et multithreadée (temps d'exécution, scalabilité, goulots d'étranglement...).
  - **Deuxième partie — Activité DTMF :**
    - \* Une description de la partie de votre code qui a été parallélisée.
    - \* Une explication claire de la stratégie de parallélisation utilisée.
    - \* Une justification des choix effectués, et une évaluation de l'intérêt et de l'efficacité de votre parallélisation.

#### Information

N'hésitez pas à modifier le `CMakeLists.txt` pour générer directement deux exécutables : une version mono-threadée et une version multithreadée.