

## Zadanie nr 4

Napisz program realizujący operację sortowania z wykorzystaniem technologii OpenMP, C++ Thread Library oraz MPI.

### 1. Wymagania podstawowe na ocenę 3.0

1.1. Program realizuje operację sortowania:

- a) losowo wygenerowanych wartości znajdujących się w wierszach macierzy o rozmiarach  $m \times n$ . Wartości  $m$ ,  $n$  oraz zakres losowych wartości zadawane są jako argumenty linii poleceń.
- b) wartości znajdujących się w wierszach macierzy wczytywanych z pliku csv. Nazwa pliku wejściowego jak i nazwa pliku zawierającego posortowane wiersze zadawane są jako argumenty linii poleceń. Wymagany format pliku został podany w zadaniu nr 2 w p. 1.2.

Efektem działania algorytmu jest macierz składająca się z  $m$  wierszy zawierających  $n$  elementów  $a_{i,j}$  gdzie  $a_{i,j} < a_{i,j+1}$ .

1.2. Program realizuje sortowanie w oparciu o poprawnie zaimplementowany algorytm dając prawidłowe dane wyjściowe.

1.3. Zastosowane mechanizmy dają efektywne zrównoleglenie  $T_1 > T_P$  (przy odpowiedniej wielkości zadania).

1.4. Program uruchamia się z parametrami linii poleceń (tryb wsadowy):

```
m n min max plik_macierzy_wynikowej P
```

dla implementacji punktu 1.1a) gdzie  $m$  i  $n$  to rozmiary generowanej macierzy a  $min$  i  $max$  to zakres wartości jej elementów, oraz

```
plik_macierzy_zrodlowej plik_macierzy_wynikowej P
```

dla implementacji punktu 1.1b). Parametr  $P$  określa liczbę wątków w przypadku implementacji OpenMP oraz C++ Thread Library. W przypadku implementacji MPI dane dystrybuowane są przez proces o identyfikatorze 0 do pozostałych procesów, które ignorują argumenty linii poleceń.

1.5. Program uruchamia się i kończy bez błędów, a jego kod źródłowy zawarty jest maksymalnie w 3 plikach, które należy przesłać (niepakowane) na platformę elearningową.

### 2. Wymagania dodatkowe na ocenę $\leq 5.0$

2.1. Program realizuje sortowanie wszystkich elementów macierzy

- 2.1.1. w ramach implementacji OpenMP.
- 2.1.2. w ramach implementacji C++ Thread Library.
- 2.1.3. w ramach implementacji MPI.

Zrównolegleniu podlega proces sortowania w ramach poszczególnych wątków lub procesów jak również proces scalania. Efektem działania algorytmu jest macierz o rozmiarach  $m \times n$  zawierające elementy  $a_{i,j}$ , takie że  $a_{i,j} < a_{i,j+1}$  oraz  $a_{i,n} < a_{i+1,1}$ .

2.2. Optymalnie równomierna dystrybucja zadań pomiędzy procesy MPI, zakładając jednakową wydajność jednostek obliczeniowych.