

# Sprawozdanie z zadania MPI CZ. 2 - Przetwarzanie współbieżne

Piotr Trojan

## Cel zadania:

Celem zadania było doskonalenie umiejętności programowania w środowisku obliczeń równoległych z przesyłaniem komunikatów oraz implementacja programu, który przesyła złożoną strukturę danych między procesorami. Program miał także przeprowadzić pomiar wydajności operacji przesyłania tych struktur w systemie równoległym.

## Kroki wykonania:

### 1. Utworzenie podkatalogu roboczego:

Na początku zadania utworzyłem dedykowany podkatalog roboczy, w którym będzie przechowywany kod źródłowy, pliki wykonywalne oraz wyniki testów. Podkatalog ten znajduje się w systemie plików na dysku lokalnym.

### 2. Zaprojektowanie "bogatej" struktury danych:

Została zaprojektowana struktura danych w języku C, reprezentująca informacje o tkaninie. Struktura zawierała cztery pola:

- width (szerokość) – liczba całkowita.
- height (wysokość) – liczba całkowita.
- color (kolor) – tablica znaków, przechowująca nazwę koloru.
- material (materiał) – tablica znaków, przechowująca nazwę materiału.

Struktura ta może reprezentować na przykład dane produktu w sklepie tekstylnym lub kartę pacjenta w kontekście medycznym.

### 3. Napisanie programu równoległego:

Program został napisany w języku C z wykorzystaniem biblioteki MPI (Message Passing Interface). Celem było przesłanie zaprojektowanej struktury fabric między dwoma procesorami. Program wykorzystywał specjalnie utworzony nowy typ danych MPI, który odwzorowywał strukturę fabric w sposób dostosowany do przesyłania przez MPI. Został użyty typ MPI\_Type\_struct, który umożliwił zdefiniowanie struktury o różnych typach danych (całkowite i znaki).

Program składał się z następujących etapów:

- Zainicjowanie środowiska MPI.
- Utworzenie niestandardowego typu danych przy pomocy MPI\_Type\_struct.
- Przesyłanie danych między procesami za pomocą funkcji MPI\_Send i MPI\_Recv.
- Pomiar czasu wykonania operacji przesyłania za pomocą funkcji MPI\_Wtime.

#### 4. Uruchomienie programu i testy wydajności:

Program został uruchomiony na dwóch procesach, z których jeden wysyłał dane do drugiego. Czas wykonania przesyłania danych został zmierzony dla operacji między procesami. Wyniki pokazały bardzo krótki czas przesyłania struktury (około 0.000019 sekund dla małych danych). Testy wydajności przeprowadzono dla różnych liczby procesów i wielokrotnych przesyłaniach, aby ocenić, jak zmienia się czas operacji w zależności od liczby procesów oraz wielkości danych.

#### 5. Powtórzenie testu z użyciem spakowanego typu MPI:

Dalszym krokiem było zastosowanie spakowanego typu danych MPI (np. MPI\_Type\_pack). Spakowanie pozwala na optymalizację przesyłania danych, redukując dodatkowe informacje o strukturze, co może prowadzić do szybszego przesyłania.

## Wyniki testów:

Po uruchomieniu programu z użyciem dwóch procesów (rank 0 i rank 1), dane struktury zostały poprawnie przesłane między procesami. Na ekranie wyświetlone zostały informacje o strukturze, a także czas przesyłania:

```
ini
```

```
KopiujeDytuj
```

```
rank = 0 :
```

```
width = 13
```

```
height = 20
```

```
color = blue
```

```
material = cotton
```

```
time = 0.000019
```

Czas wykonania operacji przesyłania danych między procesami w tym przypadku wyniósł zaledwie 0.000019 sekundy, co wskazuje na bardzo szybkie przesyłanie przy małych danych.

## **Podsumowanie:**

W wyniku realizacji zadania udało się stworzyć program równoległy przesyłający złożoną strukturę danych między procesorami za pomocą biblioteki MPI. Program został przetestowany pod kątem wydajności, a wyniki wskazują na efektywność przesyłania danych w przypadku małych struktur. Dalsze eksperymenty z większą liczbą procesów oraz większymi strukturami mogą pomóc w dalszej optymalizacji i lepszym zrozumieniu wydajności obliczeń równoległych w systemach rozproszonych.

Zadanie pozwoliło na zdobycie doświadczenia w pracy z MPI oraz w optymalizacji przesyłania danych w systemach równoległych.