

**Politechnika Krakowska  
im. Tadeusza Kościuszki**

Wydział Informatyki i Telekomunikacji

**Klaudia Kromołowska**

**Piotr Gucwa**

**SPRAWOZDANIE Z PROJEKTU**

*Obliczanie średniej arytmetycznej liczb pięciocyfrowych z zadanego przedziału spełniających określone warunki*

**PROGRAMOWANIE RÓWNOLEGŁE**

**I ROZPROSZONE**

Spis treści

[Opis problemu 2](#_Toc119506714)

[Parametry maszyny 2](#_Toc119506715)

[OpenMP – część teoretyczna 2](#_Toc119506716)

[MPI – część teoretyczna 2](#_Toc119506717)

[Podejście hybrydowe – część teoretyczna 3](#_Toc119506718)

[Implementacja – generowanie danych 3](#_Toc119506719)

[Implementacja – OpenMP 3](#_Toc119506720)

[Implementacja – MPI 3](#_Toc119506721)

[Implementacja – OpenMP + MP 3](#_Toc119506722)

[Wyniki – wykresy 3](#_Toc119506723)

[Wnioski 3](#_Toc119506724)

# Opis problemu

Celem projektu jest porównanie czasów oblliczeń potrzebnych do wyliczenia średniej arytmetycznej liczb przy pewnych założeniach:

* Liczby należą do zadanego przedziału
* Liczby są pięciocyfrowe
* Suma cyfr liczby należy do zadanego przedziału.

Porównanie czasów obliczeń dotyczy trzech implementacji – z użyciem interfejsu OpenMP, interfejsu MPI oraz wersji hybrydowej (użycie jednocześnie OpenMP i MPI).

# Parametry maszyny

|  |  |
| --- | --- |
| Wersja systemu operacyjnego |  |
| Typ systemu operacyjnego |  |
| Procesor – rodzina |  |
| Procesor – seria |  |
| Liczba rdzeni |  |
| Liczba wątków |  |
| Taktowanie rdzenia |  |
| Ilość pamięci RAM |  |
| Szybkość pamięci operacyjnej |  |

# OpenMP – część teoretyczna

OpenMP (ang. *Open Multi-Processing*) jest wieloplatformowym interfejsem programowania aplikacji. Dzięki niemu możliwe jest tworzenie programów dla systemów wieloprocesowych z dzieloną pamięcią. Dostosowany jest od architektury Unix, czy Windows, można wykorzystywać go w językach C, C++. Składa się z dyrektyw kompilatora, bibliotek i zmiennych środowiskowych, za pomocą których można definiować działanie programu. Cechuje go duża prostota użycia, a także elastyczność i skalowalność

# MPI – część teoretyczna

MPI (ang. *Message Passing Interface*), tłumaczone jako „interfejs transmisji wiadomości”, to interfejs, który umożliwia przesyłanie komunikatów między różnymi procesami równoległych programów. Najczęściej implementowany jest w postaci bibliotek dla języków C i C++. Cechuje go wysoka jakość, duża skalowalność i przenośność.

# Podejście hybrydowe – część teoretyczna

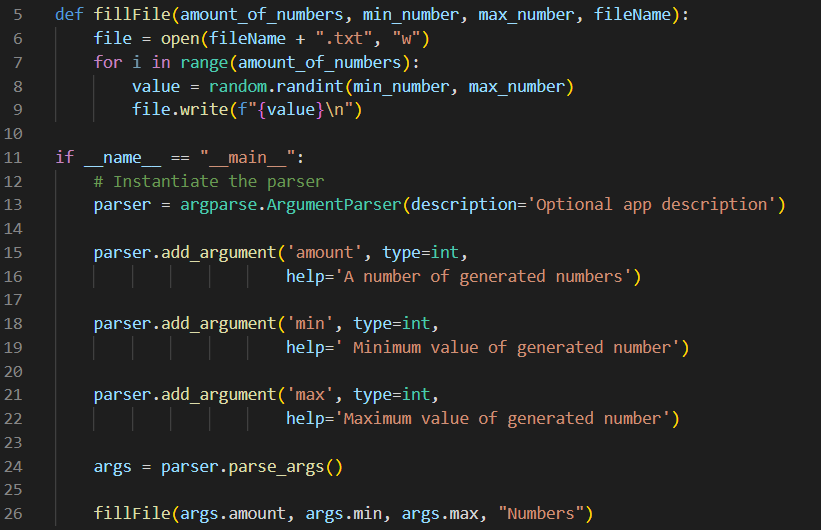
Najciekawszym podejściem jest jednak zastosowanie połączenia OpenMP oraz MPI – takie rozwiązanie hybrydowe można stosować do tworzenia aplikacji działających równolegle na wieloprocesorowych maszynach. Programy uruchamiane są na klastrach pod kontrolą MPI, a do zrównoleglenia pracy węzłów klastrów wykorzystywany jest interfejs OpenMP.

# Implementacja – generowanie danych

Pierwszą częścią projektu było wykonanie algorytmu, który będzie generował odpowiednie dane. W tym celu przygotowano skrypt w języku Python – na wejście dostaje on 3 parametry:

* *amount* – ilość danych do wygenerowania,
* *min* – dolna granica przedziału w której muszą mieścić się liczby
* *max* – górna granica przedziału w której muszą mieścić się liczby.

Wszystkie wygenerowane liczby zapisywane są do pliku .txt



# Implementacja – OpenMP

# Implementacja – MPI

# Implementacja – OpenMP + MPI

# Wyniki – wykresy

# Wnioski