|  |  |
| --- | --- |
| Akademia Nauk Stosowanych w Nowym Sączu  Wydział Nauk Inżynieryjnych | |
| Imię i nazwisko: | Michał Bernardy |
| Grupa: | P1 |
| Ocena: |  |

## Zadanie 1: Zrównoleglenie obliczeń metody Monte Carlo (obliczanie PI)

**Opis algorytmu**

Metoda Monte Carlo wykorzystuje losowe punkty do przybliżonego obliczenia wartości liczby PI. Idea polega na losowym generowaniu punktów w kwadracie jednostkowym i zliczaniu, ile z nich znajduje się w ćwiartce koła wpisanego w ten kwadrat. Stosunek liczby punktów wewnątrz ćwiartki do całkowitej liczby punktów pomnożony przez 4 daje przybliżenie wartości PI.

**Zrównoleglenie**

Zastosowano zrównoleglenie z użyciem biblioteki pthread. Każdy wątek:

* losuje własny zbiór punktów,
* oblicza liczbę punktów wewnątrz ćwiartki koła,
* przekazuje wynik do struktury danych.

Po zakończeniu wszystkich wątków, główny wątek sumuje wyniki i oblicza ostateczne przybliżenie liczby PI.

**Kod źródłowy**

(*pełny kod znajduje się w załączniku – zad1.c*)

**Wyniki działania**

Dla 8 wątków i 100 milionów punktów:

* Przybliżona wartość PI: ~3.14159
* Czas wykonania: ok. 1.5 sekundy

**Zadanie 2: Równoległe mnożenie macierzy**

**Opis algorytmu**

Zaimplementowano klasyczny algorytm mnożenia macierzy A[n][m] × B[m][p] = C[n][p].

Wersja zrównoleglona polega na podziale macierzy wynikowej C na bloki wierszy, które są obliczane niezależnie przez poszczególne wątki.

**Zrównoleglenie**

Każdy wątek:

* oblicza fragment wynikowej macierzy C,
* wykonuje klasyczne mnożenie w wyznaczonym zakresie wierszy.

Zastosowano pthread do uruchomienia 8 wątków.

**Kod źródłowy**

(*pełny kod znajduje się w załączniku – zad2.c*)

**Wyniki działania**

Dla macierzy o wymiarach 1000×1000:

* Czas wykonania: ok. 2.8 sekundy
* Liczba wątków: 8

**Wnioski**

* Zrównoleglenie metody Monte Carlo przynosi wymierne korzyści czasowe przy bardzo dużych liczbach prób.
* Mnożenie macierzy jest naturalnie równoległym problemem i również bardzo dobrze skalowalne.
* Oba algorytmy zostały poprawnie zrównoleglone i uzyskano istotne przyspieszenie względem potencjalnych wersji sekwencyjnych.