Systemy równoległe i rozproszone

Aplikacja rozproszona

Mnożenie macierzy - algorytm Cannon’a

Dawid Sroka
Jakub Smuga

1. Wstęp

W dzisiejszym świecie obliczeń inżynieryjnych, efektywność operacji na macierzach odgrywa kluczową rolę. Jest jedną z fundamentalnych operacji i jest wykorzystywana w szerokim zakresie w bardzo różnorodnych dziedzinach, począwszy od analizy danych po obliczenia graficzne.
Algorytm Cannon’a cechuje się równomiernym rozłożeniem pracy pomiędzy procesory oraz minimalną komunikacją między nimi, co sprawia, że jest szczególnie efektywny w przypadku dużych macierzy i architektur równoległych.

2. Algorytm Cannon’a

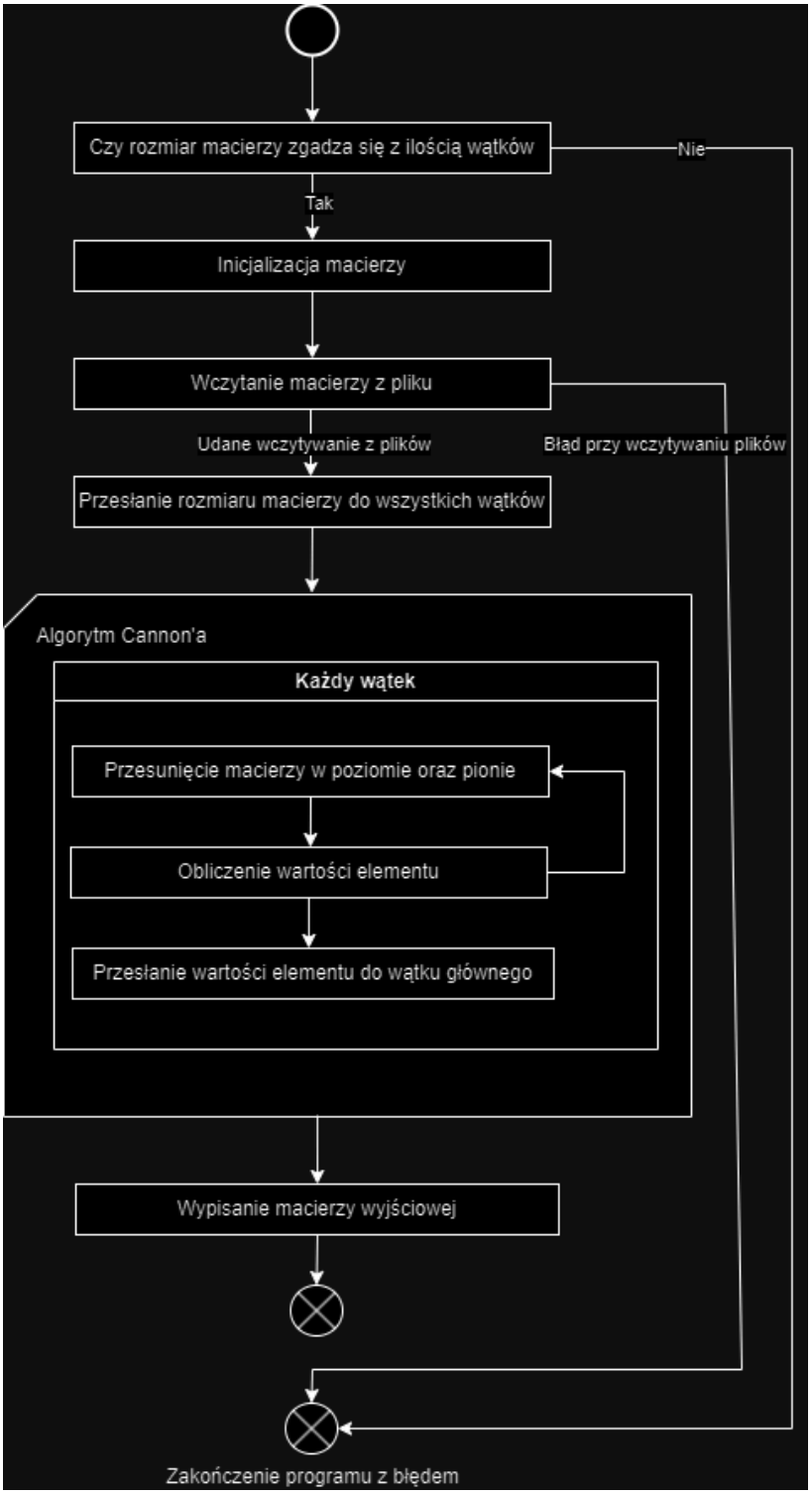
Algorytm Cannon'a można zapisać sekwencją kolejnych króków:

a. **Podział macierzy na bloki**: Pierwszym krokiem algorytmu jest podział każdej z macierzy wejściowych na bloki o jednakowych rozmiarach. Bloki te są przypisywane do odpowiednich procesorów w siatce równoległej.
b. **Przesunięcie bloków macierzy**: Kolejnym etapem jest przesunięcie bloków macierzy wzdłuż odpowiednich wymiarów macierzy. Każdy procesor przemieszcza swoje bloki do sąsiadujących procesorów zgodnie z ustalonym kierunkiem cyklicznym.
c. **Mnożenie lokalne**: Na każdym procesorze następuje lokalne mnożenie bloków macierzy, które zostały przypisane do tego procesora. Każdy procesor wykonuje operację mnożenia macierzy lokalnie.
d. **Agregacja wyników**: Wyniki lokalnego mnożenia są agregowane poprzez cykliczne przesunięcia bloków macierzy. Proces ten pozwala na skumulowanie częściowych wyników mnożenia z różnych procesorów.

e. **Powtarzanie cykli**: Kroki 2-4 są powtarzane określoną liczbę razy, tak aby każdy blok macierzy był przetwarzany przez każdy procesor w siatce równoległej. To pozwala na równomierny rozkład obciążenia pracy pomiędzy procesory.

3. Struktura programu

a. Schemat blokowy



b. Struktury danych
    - **matrix_a** - Pierwsza wejściowa macierz
    - **matrix_b** - Druga wejściowa macierz
    - **matrix_c** - Macierz wyjściowa

c. Funkcje oraz operacje
    - **shift_matrix_horizontal** - Funkcja pomocnicza do przesunięcia macierzy o k kroków w poziomie
    - **shift_matrix_vertical** - Funkcja pomocnicza do przesunięcia macierzy o k kroków w pionie
    - **cannon_algorithm** - Główna funkcja mnożenia, używająca pomocniczych funkcji do przesuwania macierzy
    - **main** - Funkcja wejściowa wczytująca macierze i inicjalizująca macierze, wywołująca inne funkcje oraz wypisująca macierze

d. Ograniczenia i założenia
    - Liczba wątków musi być równa ilości elementów w macierzy
    - Macierz musi być kwadratowa
    - Elementy macierzy muszą być liczbami całkowitymi

e. Uruchomienie
    - Kompilacja programu

```
mpicc main.cpp -o main
```

    - Uruchomienie programu

```
mpiexec -n X ./main <NAZWA_1_PLIKU_Z_MACIERZA>
<NAZWA_2_PLIKU_Z_MACIERZA> <X>
```

X - liczba wątków oraz rozmiar macierzy

Przykładowe uruchomienie programu w pracowni 204

```
source /opt/nfs/config/source_mpich420.sh &&
source /opt/nfs/config/source_cuda121.sh &&
/opt/nfs/config/station204_name_list.sh 1 16 > nodes &&
mpicc main.cpp -o main &&
mpiexec -f nodes -n 225 ./main matrix_A15.txt matrix_B15.txt 15
```