

Sprawozdanie Przetwarzanie równoległe Projekt 1 OMP

Dominik Doberski, Artur Olejnik

23 maja 2019

1 Wstęp

1.1 Temat

Mnożenie macierzy - porównanie efektywności metod –

- 3 pętle - kolejność pętli: jik, podział pracy przed pętlą 1
- 6 pętli -kolejność pętli: zewnętrznych ijk, wewnętrznych: ii,kk,jj podział pracy przed pętlą 2.

1.2 Autorzy

Dominik Doberski	132207	dominik.doberski@student.put.poznan.pl
Artur Olejnik	122402	artur.olejnik@student.put.poznan.pl

Grupa dziekańska I3

Termin zajęć: poniedziałek 16:50

1.3 Opis systemu obliczeniowego

Do wykonania pomiarów wykorzystano program CodeXL. Testy zrealizowano na jednostce obliczeniowej o poniższej specyfikacji:

- Nazwa: AMD FX-6300 Black Edition
- Częstotliwość taktowania: 3.5 GHz (max 3.8 GHz)
- Liczba procesorów fizycznych: 6
- Liczba procesorów logicznych: 6

- Liczba uruchamianych w systemie wątków: 6
- Wielkość i organizacja pamięci podręcznych procesora:
L1 dane: 6x16KB
L1 instrukcje: 3x64KB
L2: 3x2MB
L3: 8MB

2 Analiza algorytmu

2.1 Kluczowe fragmenty kodu

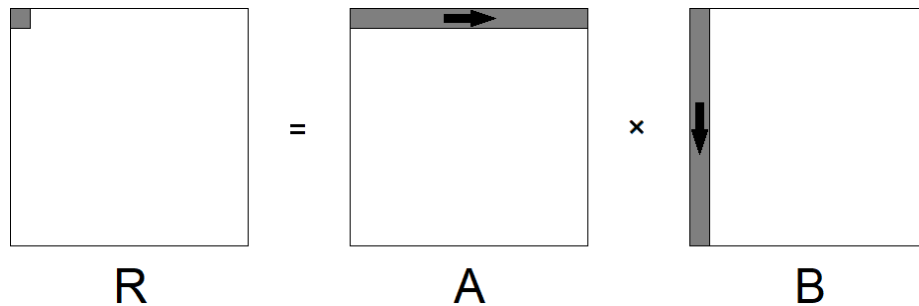
Funkcja 1: 3 pętle – jik

```

1 void multiply_matrices_JIK() {
2 #pragma omp parallel
3 #pragma omp for
4   for (int j = 0; j < ROWS; j++){
5     for (int i = 0; i < ROWS; i++){
6       for (int k = 0; k < ROWS; k++){
7         matrix_r[i][j] += matrix_a[i][k] * matrix_b[k][j];
8       }
8     }
8   }

```

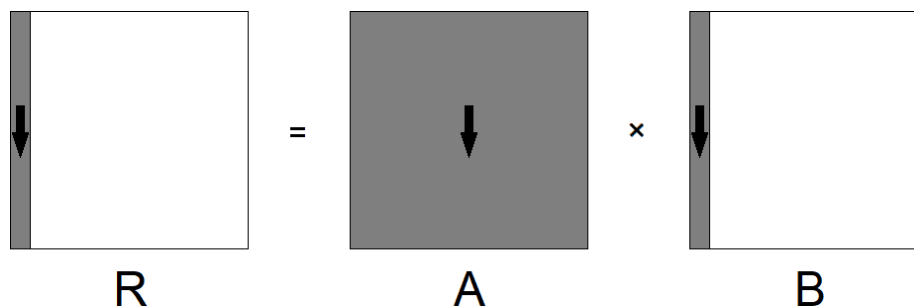
W powyższej funkcji pętla w 6. wierszu iteruje po kolumnach macierzy A i po wierszach macierzy B . Iloczyn tych dwóch elementów jest dodawany do pojedynczego elementu macierzy R znajdującego się w i -tym wierszu i j -tej kolumnie, które są wyznaczane w pętlach odpowiednio 5. i 4. wierszu. Złożoność algorytmu wynosi $O(n^3)$.



Rysunek 1: Zasoby wykorzystywane w pętli wewnętrznej

Powyższy rysunek przedstawia dokładnie, które dane są wykorzystywane do przetworzenia pętli w 6. linii kodu. Jak widać do wykonania jednej iteracji pętli wewnętrznej potrzebne jest odczytanie jednego wiersza macierzy A i jednej

kolumny macierzy B. Rezultatem tej pętli będzie odczytanie jednej wartości macierzy wynikowej R.



Rysunek 2: Zasoby wykorzystywane w pętli środkowej

Drugi rysunek ilustruje wykorzystanie danych do zrealizowania pętli w 5. wierszu. W tym przypadku potrzeba już wszystkich danych z macierzy A i, tak jak w przypadku samej pętli wewnętrznej, jednej kolumny macierzy B. Otrzymujemy całą kolumnę macierzy wynikowej R. Wykonanie pętli zewnętrznej pozwoli na uzyskanie wartości wszystkich kolumn macierzy R.

Dyrektywy OpenMP w 2. i 3. wierszu wykorzystano do uruchomienia kodu na wielu wątkach oraz podzielenia przetwarzania pętli między nimi. Kolejne iteracje pętli w 4. wierszu zostaną przydzielone równomiernie między wątki. Spowoduje to, że kolumny macierzy B i macierzy wynikowej R nie będą się powtarzały w kolejnych wątkach.

Funkcja 2: 6 pętli – zewnętrzne ijk, wewnętrzne ii, kk, jj

```

9 void multiply_matrices_6(int r){
10 #pragma omp parallel
11 for (int i = 0; i < ROWS; i += r) {
12 #pragma omp for
13 for (int j = 0; j < ROWS; j += r) {
14 for (int k = 0; k < ROWS; k += r) {
15 for (int ii = i; ii < i + r; ii++) {
16 for (int kk = k; kk < k + r; kk++) {
17 for (int jj = j; jj < j + r; jj++) {
18 matrix_r[ii][jj] += matrix_a[ii][kk] * matrix_b[kk][jj];
19 }
16 }
15 }
14 }
13 }
12 }
11 }

```

Druga funkcja jest bardziej złożona. 3 wewnętrzne pętله pracują podobnie jak pętله w poprzedniej funkcji, ale pracują tylko dla określonego fragmentu każdej z tych macierzy. Wielkość tego obszaru jest określona w argumencie r . Pętله w 14. wierszu sumuje wartości komórek otrzymanego fragmentu z iloczynami innych potrzebnych fragmentów. Pętله w 11. i 13. wierszu iterują po każdym fragmencie w celu wyliczenia wartości każdej komórki macierzy wynikowej R. Złożoność tego rozwiązania wynosi $O((\frac{n}{r})^3 * r^3) = O(n^3)$.

Dyrektywy OpenMP działają podobnie jak w pierwszej funkcji. Tutaj pomiędzy wątki równomiernie zostają rozdzielone kolejne iteracje pętli w 13. wierszu.

2.2 Analiza efektywności

2.2.1 Podział pracy na wątki

W pierwszej funkcji wątek dokonuje obliczeń dla pojedynczej kolumny macierzy wynikowej R . Wykorzystuje do tego jedną kolumnę macierzy B i całą macierz B . Dla drugiej funkcji zrównoleglenie odbywa się dla pętli w 13. wierszu, a więc wątki również dokonują obliczeń dla kolumn macierzy wynikowej R . Różnica polega na tym, że w tym przypadku obliczenia są dokonywane nie dla jednej kolumny, a dla grupy r kolumn, gdzie r jest argumentem tej funkcji.

2.2.2 False sharing

Z powyższych rozważań wynika, że w obu funkcjach każdy wątek w jednej chwili oblicza inny fragment macierzy wynikowej R . Nie będzie zatem potrzebne synchronizowanie pracy różnych wątków i nie dojdzie do unieważnienia danych – false sharing nie zachodzi.

2.2.3 Analiza lokalności dostępu do danych

Rozróżniamy dwa rodzaje lokalności – czasową i przestrzenną. Pierwsza oznacza wielokrotne odwoływanie się wątków do danych przetrzymywanych w pamięci podręcznej. Do jej wystąpienia wymagane jest odpowiednio duży rozmiar pamięci podręcznej, która musi być większa od rozmiaru pamięci potrzebnej do wykonania danej operacji. Lokalność przestrzenna polega na dostępie do danych występujących jeden po drugim w pobranej linii pamięci czyli w jednym wierszu macierzy.

W poniższych punktach przeprowadziliśmy analizę pod względem występowania lokalności dla różnych rozmieszczeń dyrektywy `#pragma omp for`.

a) Lokalność dla pierwszej funkcji

- Przed pętlą w 4. linii kodu
Powyższy rysunek przedstawia, które elementy każdej z 3 macierzy są brane pod uwagę w pętli zewnętrznej.
- Przed pętlą w 5. linii kodu

b) Lokalność dla drugiej funkcji

3 Analiza przetwarzania

3.1 Przebieg

asdf

3.2 Wyniki

asdf

3.3 Prezentacja wzorców

asdf

3.4 Miary przetwarzania

asdf

4 Wnioski

Chujowy projekt.

4.1 Wypunktowania

To jest wypunktowanie:

- punkt
- punkt
- punkt

4.2 Wyliczenie

To jest wyliczenie:

1. pierwszy,
2. drugi,
3. trzeci.

4.3 Wykres

Na Rysunku ?? znajduje się przykładowy wykres.

4.4 Wzory

$$2 + 2 = 4 \tag{1}$$

$$E = mc^2 \tag{2}$$

$$\left[-\frac{\hbar^2}{2M} + V(X) \right] \Psi(x) = \mathcal{E} \Psi(x) \tag{3}$$

5 Podsumowanie