do zdobycia: [30pkt]
czas realizacji: 3 tygodnie

Zestaw 4

Celem ćwiczenia jest napisanie programu równoległego służącego do przetwarzania macierzy. Powstały program ma obliczać macierz $C = AB$, tj. macierz będącą iloczynem dwóch macierzy A i B. W przypadku gdy macierze A i B posiadają (odpowiednio) wymiary $p \times q$ oraz $q \times r$ (podano liczbę kolumn i wierszy), to macierz C – będąca iloczynem A i B – posiada wymiary $p \times r$, zaś wartości jej elementów dane są wzorem:

$$c_{ij} = \sum_{k=1}^{q} a_{ik} b_{kj}. \tag{1}$$

Element c_{ij} macierzy C oblicza się zatem jako sumę iloczynów elementów i-tego wiersza macierzy A oraz j-tej kolumny macierzy B. Fakt ten stanowi istotną obserwację w kontekście potencjalnego zrównoleglenia operacji mnożenia macierzy.

Problem obliczania iloczynu dwóch macierzy można zdekomponować ze względu na wynik, tj. macierz C. Każdemu z n procesów można przyporządkować podmacierz macierzy wynikowej C, powierzając mu (jako pracę do wykonania) obliczenie wartości wszystkich elementów przynależnej mu podmacierzy. Zastosowanie takiego podejścia wiąże się z koniecznością podzielenia macierzy C na n podmacierzy. Aby dekompozycja była optymalna należy zadbać, by rozmiary podmacierzy (przez rozmiar podmacierzy rozumie się tutaj liczbę jej elementów) przypisanych poszczególnym procesom były możliwie jak najbardziej zbliżone (gwarantuje to równomierny podział pracy). Chęć ograniczenia narzutów wynikających z zastosowania przetwarzania równoległego (komunikacja), skłania do zastosowania podejścia, w którym dekomponując problem macierz C dzieli się na macierz klatkową.

Omówiony powyżej sposób dekompozycji problemu można zobrazować w sposób następujący:

$$A \cdot B = \left[\begin{array}{cc} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ \hline a_{3,1} & a_{3,2} \\ a_{4,1} & a_{4,2} \end{array}\right] \cdot \left[\begin{array}{cc|cc} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} \end{array}\right] = \left[\begin{array}{cc|cc} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ \hline c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{array}\right] = C \tag{2}$$

W przedstawionym przykładzie mnożone są dwie macierze A i B, o wymiarach wynoszących odpowiednio 4×2 oraz 2×4. Macierz wynikowa C ma wymiary 4×4. W obliczeniach wykorzystywane są 4 procesy. Każdy z procesów realizuje obliczenia dla podmacierzy składającej się z 2×2 elementów, obliczając wartości następujących elementów:

proces 0: elementy c_{ij} o indeksach $i = 1, 2$ oraz $j = 1, 2$,
tj. górny lewy fragment macierzy C (kolor fioletowy),

proces 1: elementy c_{ij} o indeksach $i = 1, 2$ oraz $j = 3, 4$,
tj. górny prawy fragment macierzy C (kolor pomarańczowy),

proces 2: elementy c_{ij} o indeksach $i = 3, 4$ oraz $j = 1, 2$,
tj. dolny lewy fragment macierzy C, (kolor cyjan)

proces 3: elementy c_{ij} o indeksach $i = 3, 4$ oraz $j = 3, 4$,
tj. dolny prawy fragment macierzy C (kolor limonkowy).

Należy tutaj zauważyć, że aby dany proces mógł przeprowadzić obliczenia, musi posiadać informację o odpowiednich elementach macierzy A i B. W przedstawionym przykładzie poszczególne procesy muszą znać wartości następujących elementów:

proces 0: elementy a_{ij} o $i = 1, 2$ oraz $j = 1, 2$ (tj. pierwsze dwa wiersze macierzy A, kolor czerwony), elementy b_{ij} o $i = 1, 2$ oraz $j = 1, 2$ (tj. pierwsze dwie kolumny macierzy B, kolor niebieski),

proces 1: elementy a_{ij} o $i = 1, 2$ oraz $j = 1, 2$ (tj. pierwsze dwa wiersze macierzy A, kolor czerwony), elementy b_{ij} o $i = 1, 2$ oraz $j = 3, 4$ (tj. ostatnie dwie kolumny macierzy B, kolor żółty),

proces 2: elementy a_{ij} o $i = 3, 4$ oraz $j = 1, 2$ (tj. ostatnie dwa wiersze macierzy A, kolor zielony), elementy b_{ij} o $i = 1, 2$ oraz $j = 1, 2$ (tj. pierwsze dwie kolumny macierzy B, kolor niebieski),

proces 3: elementy a_{ij} o $i = 3, 4$ oraz $j = 1, 2$ (tj. ostatnie dwa wiersze macierzy A, kolor zielony), elementy b_{ij} o $i = 1, 2$ oraz $j = 3, 4$ (tj. ostatnie dwie kolumny macierzy B, kolor żółty).

W ogólności, jeżeli proces m-ty ma obliczyć wartości elementów c_{ij} o $i_m^{\min} \leqslant i \leqslant i_m^{\max}$ oraz $j_m^{\min} \leqslant j \leqslant j_m^{\max}$, to musi posiadać informacje o:

1. elementach a_{ij} o indeksach $i_m^{\min} \leqslant i \leqslant i_m^{\max}$ oraz $1 \leqslant j \leqslant q$,
2. elementach b_{ij} o indeksach $1 \leqslant i \leqslant q$ oraz $j_m^{\min} \leqslant j \leqslant j_m^{\max}$.

Powstały program powinien działać w sposób opisany poniżej.

1. Program powinien pobierać z wektora argumentów 3 parametry, informujące o nazwach plików zawierających dane o macierzy A (pierwszy parametr) i macierzy B (drugi parametr). Ostatni, trzeci parametr informuje o nazwie pliku wyjściowego, do którego ma zostać zapisana obliczona macierz C.
2. Na początku następuje wczytanie informacji o macierzach A i B. Informacja ta czytana jest w sposób szeregowy, jedynie przez nadzorce (tj. proces o randze 0). Macierze A i B wczytywane są z plików (ich format został przedstawiony dalej).
3. Nadzorca znając wymiary macierzy A i B, sprawdza czy są one odpowiednie, tj. czy liczba kolumn macierzy A równa jest liczbie wierszy macierzy B. Jeżeli tak nie jest, wyświetlany jest komunikat o błędzie i program zostaje zakończony.
4. W kroku kolejnym nadzorca wyznacza stosowaną dekompozycję. Na wstępie sprawdza czy przy zadanym n (liczba procesów) możliwe jest podzielenie macierzy C (posiadającej wymiary $p \times r$) na $n_h \times n_v$ klatek o równych wymiarach $a \times b$. W kroku tym znając wartości parametrów n, p i r należy wyznaczyć cztery liczby całkowite: n_h, n_v, a oraz b. Muszą one spełniać następujące relacje:

$$n_h \times n_v = n, \tag{3}$$

$$n_h \times a = p \tag{4}$$

oraz

$$n_v \times b = r. \tag{5}$$

Jeżeli dla zadanych n, p i r nie jest możliwe wyznaczenie liczb n_h, n_v, a i b spełniających wszystkie trzy warunki (równania (3), (4) i (5)) to wyświetlany jest komunikat o błędzie i zakończone zostaje działanie programu. Jeżeli dla zadanych n, p i r istnieje większa liczba rozwiązań, to wybierane jest rozwiązanie optymalne, minimalizujące liczbę koniecznych do wysłania elementów. Jeżeli istnieją dwa (lub więcej) rozwiązań równie optymalnych, to wybierane jest którekolwiek z nich.

5. Po określeniu sposobu dekompozycji problemu, nadzorca wyświetla na ekranie informacje o wartości parametrów n, p i r oraz n_h, n_v, a i b. Na ekranie wyświetlona zostaje także informacja o przypisanych poszczególnym procesom klatkom, tj. wypisane zostają wartości parametrów $i_m^{\min}$, $i_m^{\max}$, $j_m^{\min}$ oraz $j_m^{\max}$, dla $m = 0, 1, \ldots, n-1$. Nadzorca powinien również wyświetlić na ekranie informacje o całkowitym rozmiarze danych, które przy zastosowanej dekompozycji zostaną przesłane. Rozmiar ten powinien być wyrażony w bajtach.

6. W kroku kolejnym następuje rozesłanie informacji o macierzach A i B. Do każdego z procesów przesyłana jest minimalna ilość informacji: przesyłane są jedynie wartości tych elementów macierzy A i B, które są niezbędne do obliczenia wartości elementów macierzy C przynależnej danemu procesowi klatki.

7. W następnym etapie, po otrzymaniu niezbędnych informacji, każdy z procesów oblicza wartości elementów macierzy C, realizując obliczenia jedynie dla elementów przynależnej mu klatki.

8. Po obliczeniu wartości elementów macierzy C następuje zebranie wyników. Każdy z procesów przesyła do nadzorcy informacje o obliczonej klatce macierzy C. W ostatnim etapie nadzorca porządkuje otrzymane wyniki i zapisuje macierz C do pliku.

Ponadto powstały program powinien dokonać pomiaru czasów wykonania. Po zakończeniu obliczeń nadzorca powinien zebrać i wyświetlić na ekranie informacje o czasach niezbędnych na:

1. odczytanie macierzy A i B z pliku,
2. zdekomponowanie problemu,
3. rozesłanie informacji o macierzach A i B,
4. obliczenie macierzy C,
5. zebranie wyników,
6. zapis macierzy C do pliku.

Dla każdej z powyższych kategorii powinny zostać wypisane czasy uzyskane na poszczególnych procesach oraz sumaryczny czas.

Macierze A i B (macierz C) powinny być czytane z (zapisywane do) pliku tekstowego o następującym formacie. W pierwszej linii pliku znajdują się dwie liczby całkowite, informujące o wymiarach macierzy, tj. liczba wierszy p oraz liczba kolumn q. Liczby te powinny być oddzielone jednym znakiem białym (bądź ich większą liczbą). W kolejnych p wierszach pliku zawarta jest informacja o kolejnych wierszach macierzy. Wiersz $i+1$ wiersz pliku zawiera informację o i-tym wierszu macierzy. W każdym wierszu zawarte jest q liczb zmiennoprzecinkowych, będących kolejnymi elementami i-tego wiersza macierzy. Kolejne elementy oddzielone są jednym znakiem białym (bądź ich większą liczbą). Poniżej przedstawiono postać przykładowego pliku, w którym zdefiniowana została macierz o wymiarach 6×4.

```
6 4
-0.880  0.577 -0.594 -0.303
-0.277 -0.731 -0.248 -0.481
-0.911  0.759  0.261 -0.246
-0.361  0.656 -0.150 -0.027
 0.581  0.911  0.428 -0.296
 0.684 -0.275 -0.234 -0.843
```

Inne istotne uwagi:

1. w powstałym programie proces o randze 0 (nadzorca) również powinien brać czynny udział w obliczeniach, tj. obliczać wartości elementów c_{ij} z przynależnej mu klatki,

2. nie jest dozwolone przesyłanie pełnej informacji o macierzach A i B,

3. warto w programie wyodrębnić funkcje służące do pakowania i przesyłania podmacierzy, będącej klatką zadanej macierzy,

4. przesyłając informacje o macierzach A i B warto zadbać, by liczba wykonywanych operacji komunikacji była minimalna. W najprostszym (i „zdrowym") wariancie konieczne jest wykonanie $n \times 2$ operacji wysyłania/odbioru (na każdy proces przypada jedna klatka macierzy A oraz jedna klatka macierzy B). W przypadku gdy do przesłania obu niezbędnych klatek wykorzysta się jedną operację komunikacji, liczba ta zostaje zredukowana do n operacji wysyłania/odbioru.

5. wszelka pamięć niezbędna do przechowania informacji o macierzach A, B i C powinna być alokowana dynamicznie. Należy zadbać, by program charakteryzował się minimalnymi wymaganiami pamięciowymi.