Anna Plust, 36148

Równoległe programowanie inkrementacyjne

Laboratorium 1: Zależności i transformacje

1. Pętla wykorzystana do analizy:

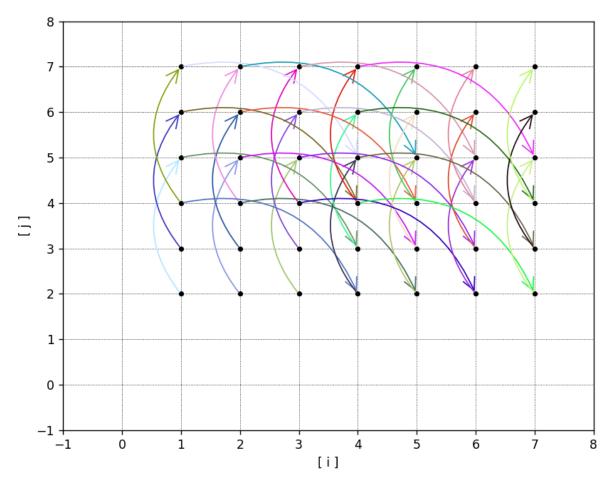
```
for (i=1; i<=n; i++) {
    for (j=2; j<=n; j++) {
        a[i][j] = a[i][j+3] + a[i+3][j-2];
    }
}</pre>
```

2. W powyższej pętli występują następujące zależności. Wypisano 6 kolejnych iteracji.

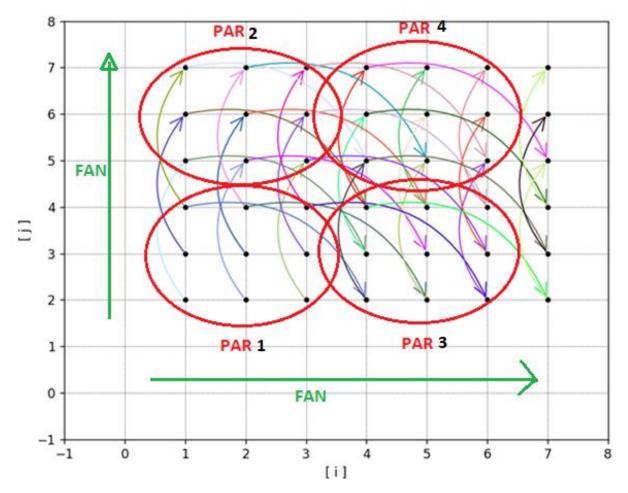
i	j	Iteracje	Zapis	Odczyt
1	2	a[1][2] = a[1][5] + a[4][0]	A[1][2]	A[1][5]
1	3	a[1][3] = a[1][6] + a[4][1]	A[1][3]	A[1][6]
1	4	a[1][4] = a[1][7] + a[4][2]	A[1][4]	A[1][7], A[4][2]
1	5	a[1][5] = a[1][8] + a[4][3]	A[1][5]	A[4][3]
1	6	a[1][6] = a[1][9] + a[4][4]	A[1][6]	A[4][4]
1	7	a[1][7] = a[1][10] + a[4][5]	A[1][7]	A[4][5]
2	2	a[2][2] = a[2][5] + a[5][0]	A[2][2]	A[2][5]
2	3	a[2][3] = a[2][6] + a[5][1]	A[2][3]	A[2][6]
2	4	a[2][4] = a[2][7] + a[5][2]	A[2][4]	A[2][7]
2	5	a[2][5] = a[2][8] + a[5][3]	A[2][5]	A[5][3]
2	6	a[2][6] = a[2][9] + a[5][4]	A[2][6]	A[5][4]
2	7	a[2][7] = a[2][10] + a[5][5]	A[2][6]	A[5][5]
3	2	a[3][2] = a[3][5] + a[6][0]	A[3][2]	A[3][5]
3	3	a[3][3] = a[3][6] + a[6][1]	A[3][3]	A[3][6]
3	4	a[3][4] = a[3][7] + a[6][2]	A[3][4]	A[3][7], A[6][2]
3	5	a[3][5] = a[3][8] + a[6][3]	A[3][5]	A[6][3]
3	6	a[3][6] = a[3][9] + a[6][4]	A[3][7]	A[6][4]
3	7	a[3][7] = a[3][10] + a[6][5]	A[3][8]	A[6][5]
4	2	a[4][2] = a[4][5] + a[7][0]	A[4][2]	A[4][5]
4	3	a[4][3] = a[4][6] + a[7][1]	A[4][3]	A[4][6]
4	4	a[4][4] = <mark>a[4][7] +</mark> a[7][2]	A[4][4]	A[4][7]
4	5	a[4][5] = a[4][8] + a[7][3]		
4	6	a[4][6] = a[4][9] + a[7][4]		
4	7	a[4][7] = a[4][10] + a[7][5]		
5	2	a[5][2] = a[5][5] + a[8][0]	A[5][2]	A[5][5]
5	3	a[5][3] = <mark>a[5][6] +</mark> a[8][1]	A[5][3]	A[5][6]
5	4	a[5][4] = a[5][7] + a[8][2]	A[5][4]	A[5][7]
5	5	a[5][5] = a[5][8] + a[8][3]		
5	6	a[5][6] = a[5][9] + a[8][4]		

5	7	a[5][7] = a[5][10] + a[8][5]		
6	2	a[6][2] = a[6][5] + a[9][0]		
6	3	a[6][3] = a[6][6] + a[9][1]	A[6][3]	A[6][6]
6	4	a[6][4] = a[6][7] + a[9][2]	A[6][4]	A[6][7]
6	5	a[6][5] = a[6][8] + a[9][3]		
6	6	a[6][6] = a[6][9] + a[9][4]		
6	7	a[6][7] = a[6][10] + a[9][5]		

3. Graf zależności:



4. Transformacje zastosowane do zrównoleglenia pętli:



Można wykorzystać transformację FAN+PAR.

PAR będzie wykorzystany dla dziewięciu podstawowych punktów zaznaczonych na czerwono (lewy dolny róg w punkcie i=1, j=2. W kolejnej iteracji PAR będzie zastosowany dla dziewięciu punktów zaczynając od i=1, j=5.

5. Pseudokod dla transformacji FAN+PAR.

W następującym pseudo kodzie można zrównoleglić obliczenia dla dziewięciu punktów.

```
for (i = 1; i < n; i = i + 3) {
    parfor (j = 2; j < n; j = j + 3) {
        parallel {
            a[i, j] = a[i][j+3] + a[i+3][j-2];
            a[i+1, j] = a[i+1][j+3] + a[i+1+3][j-2];
            a[i+2, j] = a[i+2][j+3] + a[i+2+3][j-2];
            a[i, j+1] = a[i][j+1+3] + a[i+3][j+1-2];
            a[i+1, j+1] = a[i+1][j+1+3] + a[i+1+3][j+1-2];</pre>
```

```
a[i+2, j+1] = a[i+2][j+1+3] + a[i+2+3][j+1-2];
a[i, j+2] = a[i][j+2+3] + a[i+3][j+2-2];
a[i+1, j+2] = a[i+1][j+2+3] + a[i+1+3][j+2-2];
a[i+2, j+2] = a[i+2][j+2+3] + a[i+2+3][j+2-2];
}
}
```

6. Stopień zrównoleglenia:

Po skróceniu wynik wynosi n.