

Akademia Nauk Stosowanych w Nowym Sączu		
Programowanie współbieżne i rozproszone PWIR_09		
Imię i Nazwisko: Karol Wolski	Ocena sprawozdania:	Zaliczenie:
Data: 25.04.2023	Grupa: L3	

PWIR_03_01.cpp

Zad 1.

```

Konsola debugowania programu Microsoft Visual Studio
Calculated parallel static way in 52342 milliseconds
Calculated parallel static N(MATRIX_H/10) way in 53698 milliseconds
Calculated parallel dynamic N(MATRIX_H/10) way in 53403 milliseconds
Calculated parallel guided N(MATRIX_H/10) way in 56792 milliseconds
Calculated parallel runtime way in 55676 milliseconds

C:\Users\200iq\source\repos\PWIR_09\x64\Debug\PWIR_09.exe (proces 37800) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

```

Konsola debugowania programu Microsoft Visual Studio
Calculated parallel static way in 55276 milliseconds
Calculated parallel static N(MATRIX_H/10) way in 54567 milliseconds
Calculated parallel dynamic N(MATRIX_H/10) way in 54807 milliseconds
Calculated parallel guided N(MATRIX_H/10) way in 54601 milliseconds
Calculated parallel runtime way in 54156 milliseconds

C:\Users\200iq\source\repos\PWIR_09\x64\Debug\PWIR_09.exe (proces 42968) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

```

Konsola debugowania programu Microsoft Visual Studio
Calculated parallel static way in 53958 milliseconds
Calculated parallel static N(MATRIX_H/10) way in 52921 milliseconds
Calculated parallel dynamic N(MATRIX_H/10) way in 53233 milliseconds
Calculated parallel guided N(MATRIX_H/10) way in 53273 milliseconds
Calculated parallel runtime way in 53262 milliseconds

C:\Users\200iq\source\repos\PWIR_09\x64\Debug\PWIR_09.exe (proces 2772) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

```

Konsola debugowania programu Microsoft Visual Studio
Calculated parallel static way in 52623 milliseconds
Calculated parallel static N(MATRIX_H/10) way in 53021 milliseconds
Calculated parallel dynamic N(MATRIX_H/10) way in 54271 milliseconds
Calculated parallel guided N(MATRIX_H/10) way in 54315 milliseconds
Calculated parallel runtime way in 52803 milliseconds

C:\Users\200iq\source\repos\PWIR_09\x64\Debug\PWIR_09.exe (proces 16108) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```

```
Konsola debugowania programu Microsoft Visual Studio
Calculated parallel static way in 52970 milliseconds
Calculated parallel static N(MATRIX_H/10) way in 52966 milliseconds
Calculated parallel dynamic N(MATRIX_H/10) way in 53042 milliseconds
Calculated parallel guided N(MATRIX_H/10) way in 53094 milliseconds
Calculated parallel runtime way in 53118 milliseconds

C:\Users\200iq\source\repos\PWIR_09\x64\Debug\PWIR_09.exe (proces 7492) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Średnia każdej z opcji schedule:

Static: 53 433,8

Static N(MATRIX_H/10): 53 434,6

Dynamic: 53 751,2

Guided: 54 415

Runtime: 53 803

PWIR_4_00.cpp

Zad 1

```
Konsola debugowania programu Microsoft Visual Studio
Sum calculated normal way: 4049726354 in time: 1280 ms
Sum calculated parralel way: 4049726354 in time: 1282 ms

C:\Users\200iq\source\repos\PWIR_09\x64\Debug\PWIR_09.exe (proces 24080) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Jeśli usuniemy dyrektywę reduction z kodu, wyniki sumowania będą się różnić. Bez dyrektywy reduction, każdy wątek będzie próbował aktualizować wartość zmiennej sum równocześnie, co prowadzi do sytuacji, gdy wiele wątków próbuje jednocześnie zapisywać do tej samej zmiennej.

Zad 2

```

1  #include <iostream>
2  #include <vector>
3  #include <cstdlib>
4  #include <ctime>
5  #include <cmath>
6  #include <chrono>
7
8  #define VECTOR_SIZE 10000
9
10 std::vector<int> createRandomVector()
11 {
12     std::vector<int> vec(VECTOR_SIZE);
13
14     std::srand(std::time(nullptr));
15
16     for (int i = 0; i < VECTOR_SIZE; ++i)
17     {
18         vec[i] = std::rand() % 11;
19     }
20
21     return vec;
22 }
23
24 int calculateVectorLength(const std::vector<int>& vec)
25 {
26     int sum = 0;
27
28     for (int i = 0; i < VECTOR_SIZE; ++i)
29     {
30         sum += vec[i] * vec[i];
31     }
32
33     return std::sqrt(sum);
34 }
35
36 int main()
37 {
38     std::vector<int> vec = createRandomVector();
39
40     auto start = std::chrono::high_resolution_clock::now();
41     int length = calculateVectorLength(vec);
42     auto end = std::chrono::high_resolution_clock::now();
43     std::chrono::duration<double, std::milli> durationSeq = end - start;
44
45     std::cout << "Długość wektora (wersja sekwencyjna): " << length << std::endl;
46     std::cout << "Czas wykonania (wersja sekwencyjna): " << durationSeq.count() << " ms" << std::endl;
47
48     int lengthParallel = 0;
49
50     start = std::chrono::high_resolution_clock::now();
51
52 #pragma omp parallel for reduction(+: lengthParallel)
53     for (int i = 0; i < VECTOR_SIZE; ++i)
54     {
55         lengthParallel += vec[i] * vec[i];
56     }
57
58     lengthParallel = std::sqrt(lengthParallel);
59
60     end = std::chrono::high_resolution_clock::now();
61     std::chrono::duration<double, std::milli> durationParallel = end - start;
62
63     std::cout << "Długość wektora (wersja zrownoleglona): " << lengthParallel << std::endl;
64     std::cout << "Czas wykonania (wersja zrownoleglona): " << durationParallel.count() << " ms" << std::endl;
65
66     return 0;
67 }
68

```

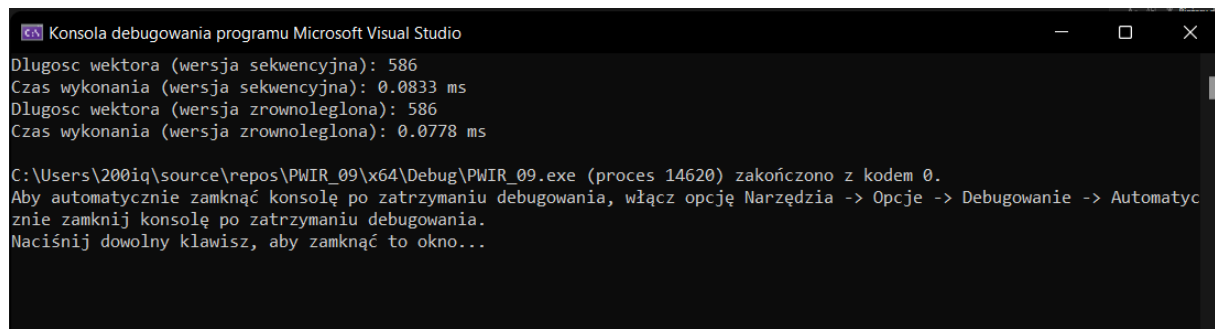
Konsola debugowania programu Microsoft Visual Studio

```

Długość wektora (wersja sekwencyjna): 593
Czas wykonania (wersja sekwencyjna): 0.177 ms
Długość wektora (wersja zrownoleglona): 593
Czas wykonania (wersja zrownoleglona): 0.1418 ms

C:\Users\200iq\source\repos\PWIR_09\x64\Debug\PWIR_09.exe (proces 14092) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...

```



```
Konsola debugowania programu Microsoft Visual Studio

Długość wektora (wersja sekwencyjna): 586
Czas wykonania (wersja sekwencyjna): 0.0833 ms
Długość wektora (wersja zrównoleglona): 586
Czas wykonania (wersja zrównoleglona): 0.0778 ms

C:\Users\200iq\source\repos\PWIR_09\x64\Debug\PWIR_09.exe (proces 14620) zakończono z kodem 0.
Aby automatycznie zamknąć konsolę po zatrzymaniu debugowania, włącz opcję Narzędzia -> Opcje -> Debugowanie -> Automatycznie zamknij konsolę po zatrzymaniu debugowania.
Naciśnij dowolny klawisz, aby zamknąć to okno...
```

Jak widzimy w powyższym przykładzie, funkcja obliczająca długość wektora ze zrównolegleniem jest szybsza od funkcji wykonującej dokładnie te same obliczenia, ale bez zrównoleglenia.