

Zadanie 4 - Liczby pierwsze - OpenMP

Celem zadania było napisanie programu, który testuje podane duże liczby pierwsze. Do poprawnego działania programu należy podać dwa argumenty wejściowe są to *n* - liczba wątków i *primes* - ścieżka do pliku z liczbami pierwszymi. Poniżej zaprezentowano główną funkcję programu odpowiedzialną za zrównoleglanie obliczeń.

```
1 int prime_number(string tnumber, int numOfThreads)
2 {
3     int prime = 1;
4     long long j;
5
6     long long number = atoll(tnumber.c_str());
7     bool flag = false;
8
9 #pragma omp parallel for shared(number, prime, flag) num_threads(numOfThreads)
10    private(j) schedule(guided, 1)
11    for (j = 2; j < number; j++)
12    {
13        if (flag)
14            continue;
15
16        if (number % j == 0)
17        {
18            prime = 0;
19            flag = true;
20        }
21    }
22    return prime;
23 }
```

Powyższy program sprawdza czy liczba jest pierwszą czy nie jest liczbą pierwszą. Do zrównoleglenia głównej pętli for została użyta dyrektywa biblioteki OpenMP. Zrównoleglenie głównej pętli for:

```
1 #pragma omp parallel for
```

Wyraz omp jest słowem kluczowym OpenMP. Dyrektywa parallel, wskazuje kompilatorowi obszar kodu, który będzie zrównoleglony. Kolejna dyrektywa - for - informuje kompilator, że zrównoleglana będzie pętla typu for. Określić które zmienne będą wspólne shared, a które prywatne private:

```
1 shared(number, prime, flag)
2 private(j)
```

Zmienne wspólne są dostępne dla każdego wątku, natomiast do danej zmiennej prywatnej ma dostęp tylko jeden określony wątek. W naszym programie zmienną prywatną jest j - licznik pętli, a zmiennymi wspólnymi number, prime, flag.

Dyrektywa schedule:

```
1 schedule(guided, 1)
```

Za pomocą polecenia schedule możemy kontrolować sposób w jaki OpenMP przydziela iteracje dostępnym wątkom. Parametr guided powoduje przypisanie do każdego wątku odpowiednio dużego fragmentu następujących po sobie iteracji. Rozmiar fragmentu zmniejsza się wykładniczo, wraz z każdym pomyślnym przypisaniem, do minimalnego rozmiaru zdefiniowanego w

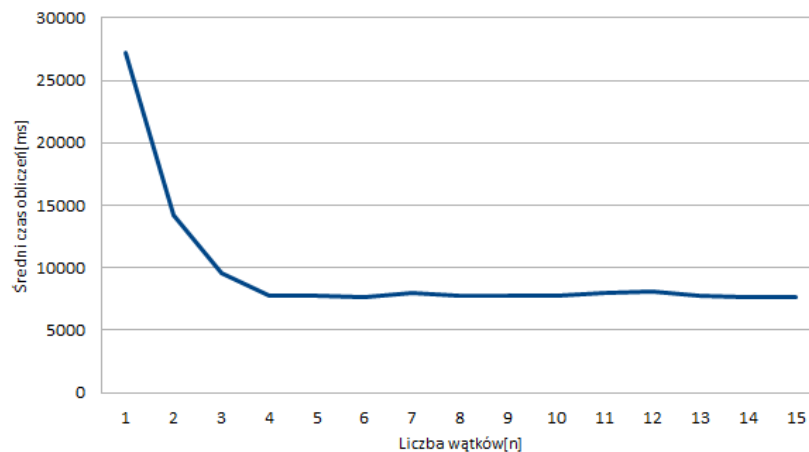
parametrze „chunk”. Parametr „chunk” jest ustawiony na 1, oznacza to że rozmiar każdego z początkowych zbiorów jest opisany wyrażeniem:

$$\text{liczba_nieprzydzielonych_iteracji} / \text{liczba_wątków}$$

Dyrektywa num_threads:

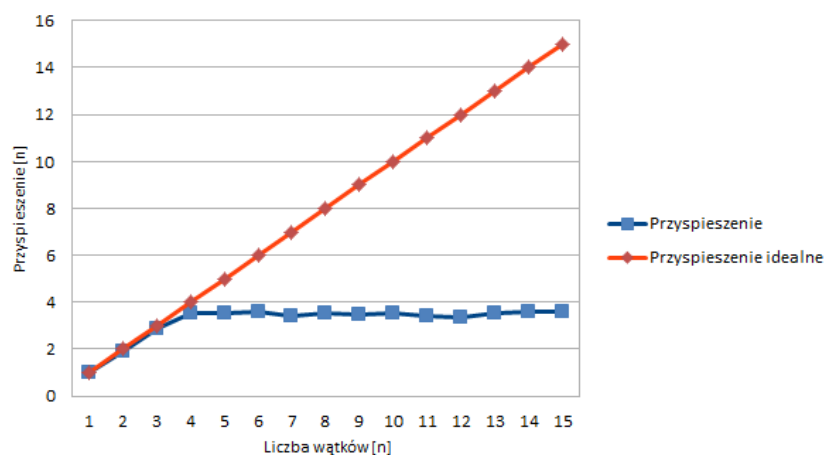
```
1 num_threads(numOfThreads)
```

Za pomocą dyrektywy num_threads określa się ile wątków ma być użytych do zrównoleglenia pętli for.



Rysunek 1: Wykres średniego czasu obliczeń

Z powyższego rysunku można wywnioskować, że średni czas obliczeń dynamicznie maleł dla pierwszych 4 wątków. Następnie dla kolejnych wątków średni czas jest ustabilizowany. Poniższy rysunek jest dobrym przykładem opisanego zjawiska.



Rysunek 2: Wykres przyspieszenia

Powyższe zadanie zostało zrównoleglone dzięki bibliotece OpenMP. Dzięki zastosowaniu dyrektyw biblioteki udało się uzyskać czterokrotne przyspieszenie dla 4 wątków. Dla kolejnych 11 wątków, w tym przykładzie nie wpłynęły znacząco na wydajność.