

## Zadanie 4 - Liczby pierwsze - OpenMP

Celem programu był test czy liczby podane w pliku są liczbami pierwszymi. W celu poprawy wydajności obliczeń program należało zrównoleglić z użyciem standardu OpenMP.

Program przyjmuje na wejściu dwa argumenty: liczbę wątków oraz ścieżkę do pliku w którym znajdują się liczby do przetestowania. Pierwszą rzeczą jaką wykonuje program jest sprawdzenie czy została podana odpowiednia ilość argumentów. Następnie zostaje wczytany plik a wpisane liczby umieszczone są w tablicy.

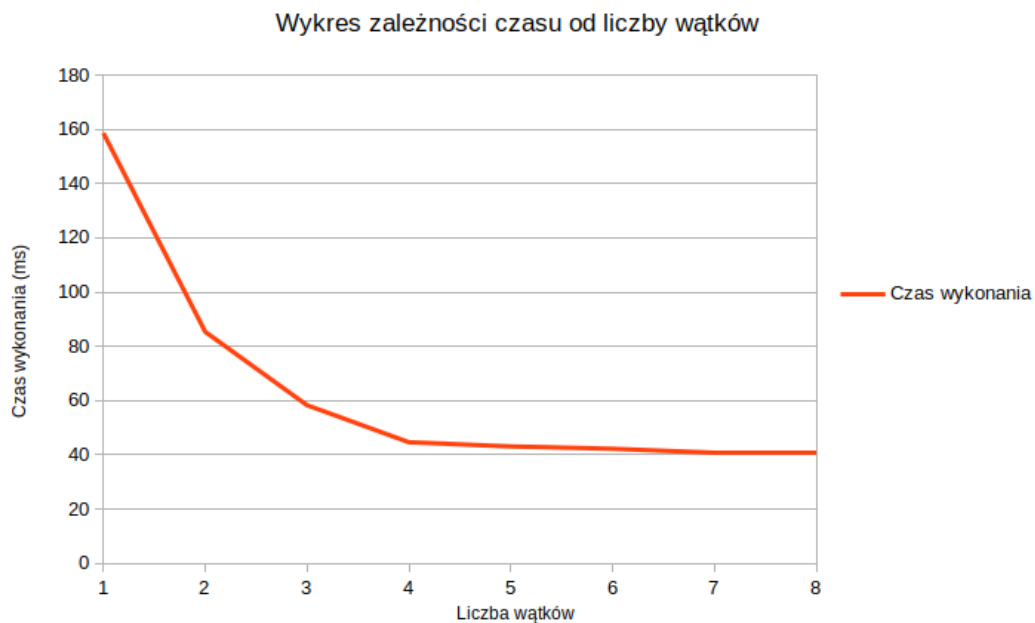
```
1  #pragma omp parallel for default(none) shared(results ,
    numbersFromFileArr) private(i) schedule(dynamic) num_threads(
    numberOfThreads)
2  for (i = 0 ; i < sizeof(numbersFromFileArr)/sizeof(*numbersFromFileArr)
    ; i++) {
3      if (isPrime(numbersFromFileArr[i])){
4          results[i] = true;
5      } else {
6          results[i] = false;
7      }
8  }
```

Powyższy fragment kodu zrównolegla działania wykonywane w pętli. Dyrektywa zrównoleglania została ustawiona na wartość default. W klauzuli shared została umieszczona tablica przechowująca wynik oszacowania czy liczba jest pierwsza oraz tablica zawierająca liczby z plików. Jako zmienna prywatna została ustalona wartość licznika pętli. Klauzula schedule została ustalona na wartość dynamic, gdyż w takiej konfiguracji udało się pozyskać najbardziej optymalne wyniki.

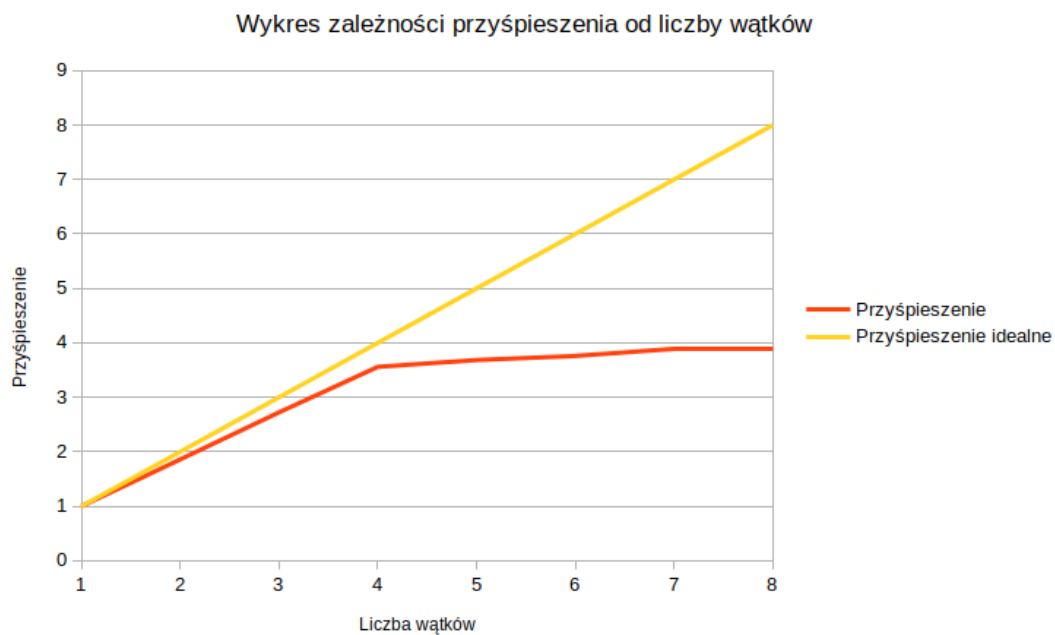
```
1 bool isPrime(int n)
2 {
3     if(n<2)
4         return false;
5
6     for(int i=2;i*i<=n;i++)
7         if(n%i==0)
8             return false;
9
10    return true;
11 }
```

Powyższy listing przedstawia algorytm wyznaczania liczby pierwszej. Został wybrany algorytm naiwny, ponieważ gwarantował poprawność wyników, a ilość i długość testowanych liczb nie była na tyle długa by powodowało to znaczące obniżenie wydajności obliczeń. W poszczególnych wątkach sprawdzane jest czy dana liczba jest liczbą pierwszą. Dane zostają podzielone na części i obliczane równoległe, używając wewnętrznych kopii danych w wątku.

Poniższe wykresy przedstawiają wyniki uzyskane na serwerze CUDA. Pierwszy wykres przedstawia zależność czasu obliczeń od liczby wątków. Drugi wykres przedstawia wartość przyspieszenia obliczeń względem programu uruchamianego synchronicznie w zależności od liczby wątków.



Rysunek 1: Wykres zależności czasu wykonywania obliczeń od liczby wątków



Rysunek 2: Wykres przyspieszenia działania programu w zależności od liczby wątków

Jak można zauważyć, program przyspiesza aż do osiągnięcia granicy 4 wątków. Dzieje się tak, gdyż procesor na którym program został uruchomiony posiada 4 rdzenie oraz korzysta z technologii „Hyper-Threading”. Dzięki wykorzystaniu technologii OpenMP udało się uzyskać przyspieszenie bliskie idealnemu, co świadczy o tym że przedstawione obliczenia dobrze nadają się do zrównoleglenia.