Grupa 1	Kordian Kurdziel, Mateusz Maciejak	07/11/2018

Zadanie 5 - Liczby pierwsze - MPI

Celem programu był test czy liczby podane w pliku są liczbami pierwszymi. W celu poprawy wydajności programu do zrównoleglenia jego działania należało wykorzystać standard MPI. Program przyjmuje dwa argumenty, obraz wejściowy oraz obraz wyjściowy. Liczbę procesów na jakich mają zostać wykoanane obliczenia należy podać po komendzie mpirun używając flagi -n.

Do komunikacji między procesami wykorzystane zostały funkcje standardu MPI - MPI_Send i MPI_Recv. Na poniższym listingu można zobaczyć ich sygnaturę.

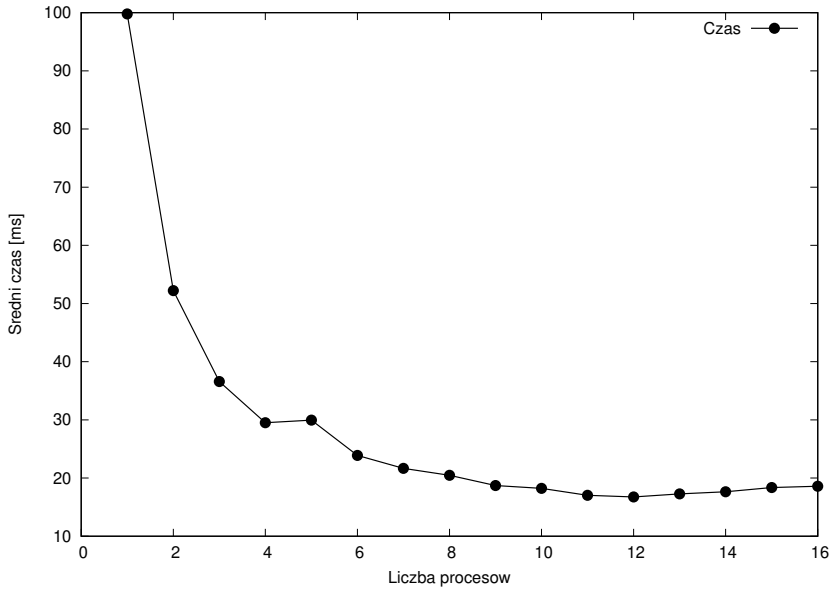
```
MPI_Send(
    void* data,
    int count,
    MPI_Datatype datatype,
    int destination,
    int tag,
    MPI_Comm communicator)
MPI_Recv(
    void* data,
    int count,
    MPI_Datatype datatype,
    int source,
    int tag,
    MPI_Comm communicator,
    MPI_Status* status)
```

Poniższy 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 równolegle, używając wewnętrznych kopii danych w procesie.

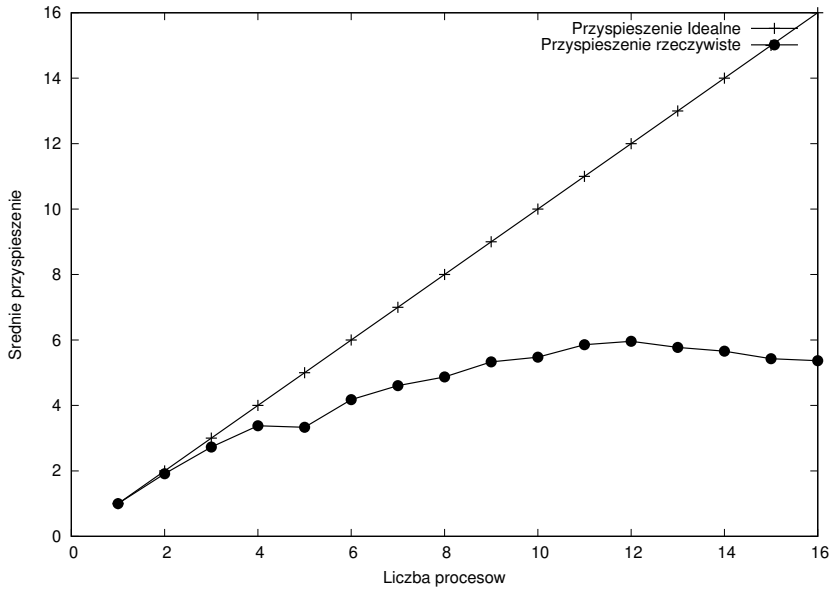
```
bool isPrime(ll n)
{
    if(n<2)
        return false;

    for(ll i=2;i*i<=n;i++)
        if(n%i==0)
            return false;

    return true;
}
```

Poniższe wykresy 1 i 2, przedstawiające zależność czasową oraz przyspieszenia zostały oparte na średnich wynikach programu uruchamianych lokalnie na maszynie wirtualnej. Wykorzystany został 6 rdzeniowy procesor Intel i7-8700K z technologią Hyperthreadingu.

Jak widać dzięki zastosowaniu technologii MPI, wykorzystując dostępne procesy, udało się znacząco przyspieszyć wykonywanie programu. Uzyskane rzeczywiste przyspieszenie jest bliskie idealnemu, co może świadczyć o tym, że powyższa klasa problemów



Rysunek 1: Wykres zależności czasu wykonywania obliczeń od liczby procesów



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

nadaje się całkiem dobrze do wykorzystywania obliczeń wielowątkowych. Na podstawie otrzymanych wyników można stwierdzić, że technologia Hyperthreadingu pomogła przyspieszyć obliczenia. Wirtualne rdzenie nie pozwalały przyspieszyć obliczenia równie sprawnie jak te rzeczywiste, ale czasy uzyskane z ich wykorzystaniem są znacząco niższe niż te bez ich użycia.