

do zdobycia: [10pkt]
deadline: 1/13 IV 2015

Zestaw 3

Tym razem naszym problemem jest zrównoleglenie operacji odczytu danych wejściowych z dysku. Jak zauważyliśmy w trakcie realizacji poprzedniego ćwiczenia, odczyt danych wejściowych stanowi w rozważanym przez nas przypadku spore (a właściwie największe) ograniczenie. Chcąc przyspieszyć działanie programu warto zadbać, by plik z danymi wejściowymi był czytany tylko raz (a nie wielokrotnie, jak to miało miejsce w naszym pierwotnym, tymczasowym, prymitywnym rozwiązaniu).

Zadaniem jest zmodyfikowanie programu z ćwiczenia poprzedniego tak, by dane wejściowe były odczytywane w sposób równoległy. Do tego celu wykorzystać należy mechanizmy dostarczane przez standard MPI, mianowicie typ `MPI_File` (czy też klasę `MPI::File`) i funkcje z nim związane (czy też metody tejże klasy).

Na wstępie przyjrzyjmy się formatowi plików wejściowych. Każda wartość (każda z trzech składowych wektora) pisana jest dokładnie na 10 polach (stosowane jest wyrównywanie do prawej i uzupełnianie polami pustymi). Dodatkowo po każdej składowej (również po ostatniej) pojawiają się 3 pola puste. Co ważne, każda z linii zakończona jest znakiem końca linii. Łatwo zatem stwierdzić iż rozmiar jednej linii to dokładnie $3 \times (10 + 3) + 1 = 40$ znaków.

Do programu dodać należy nową funkcję czytającą plik w sposób równoległy. Uruchamiając program, użytkownik powinien mieć możliwość wyboru stosowanej metody odczytu (dwie możliwości, zachować „starą” metodę odczytu, wybór za pomocą wektora argumentów). Nowa metoda odczytu powinna charakteryzować się następującymi cechami:

1. określenie zasobu danych wejściowych (tj. liczby wektorów N) dokonuje się na podstawie rozmiaru pliku wejściowego,
2. plik czytany jest dokładnie raz (każdy z procesów czyta jedynie interesujący go fragment pliku),
3. odczyt realizowany jest przy użyciu funkcji `MPI_File_read` (metody `MPI::File::Read`),
4. funkcja (metoda) odpowiedzialna za odczyt danych wołana jest na każdym procesie tylko raz, w wyniku jej wykonania odczytana zostaje całość niezbędnych danemu procesowi danych,
5. nie jest wykorzystywany mechanizm widoków,
6. dane czytane są do bufora, pamięć pod bufor jest alokowana dynamicznie, bufor jest wektorem znaków, rozmiar bufora jest minimalny,
7. po wczytaniu danych do bufora, następuje ich konwersja.

Kryteria oceniania:

1. rozsądna, prawidłowa i zgrabna metoda równoległego odczytu danych [4pkt],
2. rozsądna, prawidłowa i zgrabna metoda przechowywania i konwersji danych [2pkt],
3. estetyka i struktura kodu [2pkt],
4. analiza czasów wykonania (porównanie nowej i starej metody odczytu danych) [2pkt].