



Równoległa implementacja algorytmu sortowania bąbelkowego

W. 1.1

Wprowadzenie

Celem ćwiczenia jest opracowanie programu realizującego sortowanie bąbelkowe z wykorzystaniem zrównoleglonego algorytmu Odd-Even Sort.

UWAGA: Zarejestruj wszystkie polecenia wykonywane w trakcie realizacji zadania oraz odpowiedzi systemu; jeżeli nie jest to możliwe w sposób automatyczny – wykorzystaj mechanizm kroniki terminala SSH. Zarejestrowany materiał będzie dowodem wykonania ćwiczenia i podstawą sprawozdania. Sposób włączenia rejestracji kroniki konwersacji podany jest w materiałach kursu.

Wymagania dotyczące programu

- a) Program ma realizować sortowanie bąbelkowe ciągu liczb całkowitych z wykorzystaniem zrównoleglonej wersji algorytmu Odd-Even Sort oraz języka C + MPI.
- b) Kod źródłowy programu ma być dostarczony w pliku o nazwie `OEps.c`.
- c) Pierwszym parametrem wywołania programu jest długość N sortowanego ciągu liczb (dana całkowitoliczbowa w zakresie $<1, 10^4>$). Obecność tego parametru ma być sprawdzana.
- d) Program ma sprawdzać, czy N jest podzielne bez reszty przez liczbę p dostępnych procesów.
- e) Wartości elementów ciągu liczb do posortowania są tworzone za pomocą generatora liczb losowych o rozkładzie równomiernym na odcinku $<0, 999>$.
- f) Drugim (opcjonalnym) parametrem wywołania programu jest znak „p” lub „P” (jak *print*). Wystąpienie tego parametru oznacza, że dla celów kontrolnych mają być raportowane:
 - a. Elementy wylosowanego ciągu do sortowania,
 - b. Podciągi tego ciągu uporządkowane przez poszczególne procesy w kolejnych przebiegach,Liczby stanowiące ciąg/podciąg mają być łamane w regularne kolumny, z zachowaniem wagi pozycji poszczególnych cyfr. Żaden z tych raportów nie są produkowany, gdy długość sortowanego ciągu przekracza 20. W przypadku braku parametru „p” lub „P” raportowanie jest wyłączone.
- g) Niezależnie od obecności drugiego parametru, program ma wyprowadzić do pliku o nazwie `OEps.lst`:
 - a. Długość ciągu danych do sortowania + znak NL,
 - b. Wartości elementów do sortowania (złamane w regularne kolumny),
 - c. Znak NL,
 - d. Wartości elementów ciągu sklejonego i posortowanego (jak wyżej),
 - e. Znak NL.
- h) Program ma rejestrować czas sortowania.

Zadanie

- a) Opracuj program spełniający wymagania.
- b) Wykonaj obliczenia dla $N=10$ (z raportem kontrolnym) i $N=100$ (bez raportu kontrolnego), dla liczby procesów p nieprzekraczającej 8, ale dzielącej długość ciągu N bez reszty.
- c) Sporządź tabelaryczne zestawienie czasów sortowania dla długości ciągów i liczby procesów określonych wyżej wraz z obliczeniem przyspieszenia.

Wskazówki

Wykorzystaj dostępne poradniki programowania w języku C, w tym opis funkcji wejścia/wyjścia, a także dokumentację API Open MPI. Dodatkowe informacje zawarte są w materiałach kursu.



Opracowanie wyników

Sprawozdanie z realizacji ćwiczenia winno być wykonane na właściwym formularzu oraz zawierać m.in.

- kod źródłowy programu,
- wyniki uruchomienia programu na danych, jak opisano w punkcie **Zadanie b**).
- ocenę czasu obliczeń i przyspieszenia, jak opisano w punkcie **Zadanie c**).
- odpowiedź na pytanie, jak posortować ciąg N -elementowy na p procesorach, gdy N nie dzieli się bez reszty przez p ?

Do przesłania są 2 pliki niespakowane:

- kompletne sprawozdanie,
- plik o nazwie `OEps.c` zawierający kod źródłowy programu.