

# Parallel QuickSort

Sylwia Dzikiewicz

Marta Leniewska

Mikołaj Olszewski

Marcin Woch

Michał Żelazowski

# Podstawowe założenia

- MPI || PVM
- C++ || C
- Parametry:
  - N liczb całkowitych Int32
  - p komputerów = p procesów, gdzie  $p \cong 12$

# Ogólny schemat

- Etap 0 – sprawy organizacyjne
- Etapy 1,...,m-1
  - Przygotowanie do podziału grupy procesorów według elementu dzielącego
  - Procesor – nadzorca grupy określa podział
  - Wykonanie podziału → 2 podgrupy
- Etap m – zakończenie
  - Grupy złożone z pojedynczych procesorów
  - Sortowanie sekwencyjne elementów procesów

# Etap 0 – sprawy organizacyjne

- Proces nr. 1 (pierwszy uruchomiony) – master
- Rola mastera:
  - Przyjęcie zgłoszeń pozostałych procesów
  - Początkowe ponumerowanie pozostałych procesów
  - Wysłanie odpowiednich porcji danych do procesów 2, 3, ...,  $p$
  - Funkcja pierwszego nadzorcy (dla całości danych)

# Przygotowanie do podziału – etap k dla $k < m$

- **Wybranie przez nadzorcę** elementu dzielącego – mediana z 5 elementów
- **Wysłanie przez nadzorcę** elementu dzielącego do wszystkich procesów w swojej grupie
- **Każdy z procesów w grupie** tworzy dwa zbiory:
  - elementy mniejsze (od elementu dzielącego)
  - elementy większe lub równe
- **Każdy z procesów w grupie** informuje nadzorcę o liczności obu zbiorów (ozn.  $L_m$ ,  $L_w$ )

# Określenie podziału na etapie $k$ , dla $k < m$

- **Sortowanie przez nadzorcę** numerów procesów po liczbie elementów w zbiorach o elementach mniejszych od elementu dzielącego (czyli po  $L_m$ )
- **Wyliczenie przez nadzorcę** numeru procesu, w którym znajdzie się element dzielący
- **Wysłanie przez nadzorcę** do procesów w grupie:
  - nowego numeru procesu (kolejność po sortowaniu)
  - numerów procesów docelowych dla mniejszego zbioru danych (z uwzględnieniem liczby elementów do przesłania każdemu z nich)

# Podział na etapie k – ilustracja

- Po sortowaniu (proces reprezentuje para  $(L_m, L_w)$ ):  
(8,2) (7,3) (5,5) (3,7) (1,9)
- Element dzielący znajdzie się w procesie nr. 3, bo  
 $(8+7+5+3+1)/10 = 2.4 \cong 2$
- Przesyłanie danych (bez elementu dzielącego):

Nr. 3	5 elementów	Nr. 2
Nr. 4	3 elementy	Nr. 1
Nr. 5	1 element	Nr. 1
Nr. 2	3 elementy	Nr. 3
Nr. 1	2 elementy	Nr. 4

# Kryteria przy przesyłaniu danych...

- Jak najmniejsza ilość przesyłań
- Jak najkrótsza wiadomość
- Jak najbardziej równomierne rozłożenie elementów na procesy w obu podgrupach (procesy o numerach mniejszych od numeru procesu z elementem dzielącym, i reszta)
- Uwzględnienie konieczności przesyłania elementu dzielącego
- Wzór ???

# Podział na etapie $k$ dla $k < m$

## wykonanie

- Dla każdego procesu grupy:
  - Rozesłanie „zbędnych” elementów do odpowiednich procesów
  - Przyjęcie dodatkowych danych w sposób asynchroniczny
- Wybranie nadzorców dla 2 nowych grup – procesy o najmniejszych numerach w podgrupach
- I jeszcze raz, czyli etap  $(k+1)$

## Etap m – zakończenie

- p grup (czyli 1 grupa = 1 proces) względnie uporządkowanych
- Sortowanie elementów każdego procesu algorytmem QuickSort
- Poniżej CQS = 50 elementów zmiana sortowania na InsertionSort
- Zapisanie danych każdego procesu w pliku <nr>.txt, gdzie <nr> jest aktualnym numerem procesu

# Struktury danych

- Tablica elementów
- Lista numerów procesów
- Komunikaty

# Komunikaty

- GOTOWY
- NUMER
- DANE
- NADZORCA[numery procesów w grupie]
- LICZBA[mniejsze, większe]
- PRZEKAZ[procesy, liczby elementów]
- SORTUJ
- KONIEC

# Parallel QuickSort

**Rulez!**

