

Сортировка слиянием, параллельный алгоритм MPI

Николай Игоревич Хохлов

МФТИ, Долгопрудный

1 марта 2017 г.

Параллельная сортировка MPI

- ▶ Массив делиться на возможно равные части между процессами.
- ▶ Каждый процесс сортирует свою часть массива независимо.
- ▶ Параллельная процедура слияния отсортированных массивов в конечный массив.

Результат у процесса root, номер процесса k , число процессов p . Предполагается, что у каждого процесса уже часть массива A_k размером n_k .

Параллельная сортировка MPI: алгоритм

Algorithm 1 Сортировка MPI

```
procedure MERGESORTMPI( $A$ )  
  MergeSort( $A_k, n_k$ )  
  if  $k == \text{root}$  then  
    for  $i = 1, \dots, p - 1$  do  
      Recv  $A_i$  from  $i$   
      Создать массив  $B$  размером  $n_k + n_i$   
      Merge( $B, A_k, A_i, n_k, n_i$ )  
       $n_k \leftarrow n_k + n_i$   
       $A_k \leftarrow B$   
    end for  
     $A \leftarrow A_k$   
  else  
    Send  $A_k$  to root  
  end if  
end procedure
```

Параллельная сортировка MPI

- ▶ Сбор данных у процесса с номером root.
- ▶ Последовательный сбор данных и слияние на одном процессе.
- ▶ Последняя операция слияния выполняется последовательно.

Возможно улучшение сбора, используя алгоритмы коллективного взаимодействия по схеме All-to-one.

Параллельная сортировка MPI: алгоритм коллективного взаимодействия

Algorithm 2 Алгоритм all-to-one сортировки слиянием

MergeSort(A_k, n_k)

Пусть $p = 2^d$, $mask = 0$

for $i = 0, \dots, d - 1$ **do**

if $k \text{ AND } mask = 0$ **then**

$partner = k \text{ XOR } 2^i$

if $k \text{ AND } 2^i \neq 0$ **then**

 send A_k to partner

else

 recv A_i from partner

 Merge(B, A_k, A_i, n_k, n_i)

$n_k \leftarrow n_k + n_i$

$A_k \leftarrow B$

end if

end if

$mask = mask \text{ XOR } 2^i$

Задание

- ▶ Реализовать сортировку слиянием по схеме гиперкуб.
- ▶ Сравнить ускорение двух реализаций.
- ▶ Построить графики ускорения.