

Постановка задачи

- 1) Реализовать последовательный алгоритм сортировки "пузырьком"
- 2) Реализовать параллельный алгоритм сортировки "пузырьком" с помощью MPI
- 3) Провести эксперименты и сравнить скорость работы двух алгоритмов

Сортировка массива - расположение его элементов в определенном порядке (по возрастанию или убыванию). Она необходима для улучшения работы с большим количеством данных. Сортировка массивов облегчает поиск элементов, так как в упорядоченном массиве на него тратится намного меньше времени, а некоторые алгоритмы поиска не выполняются, если данные не отсортированы. Существует много различных методов сортировки, различающихся по эффективности.

В данной работе рассматривается алгоритм "пузырьковой" сортировки и его модификации-алгоритм "нечетно-четных перестановок" и "блочный алгоритм", позволяющие распараллелить вычисления на несколько процессоров.

Метод решения

Последовательный алгоритм пузырьковой сортировки сравнивает и обменивает соседние элементы в последовательности, которую нужно отсортировать. Для последовательности (a_1, a_2, \dots, a_n) алгоритм сначала выполняет $n-1$ базовых операций "сравнения-обмена" для последовательных пар элементов $(a_1, a_2), (a_2, a_3), \dots, (a_{n-1}, a_n)$. В результате после первой итерации алгоритма самый большой элемент перемещается ("всплывает") в конец последовательности. Далее последний элемент в преобразованной последовательности может быть исключен из рассмотрения, и описанная выше процедура применяется к оставшейся части последовательности $(a'_1, a'_2, \dots, a'_{n-1})$. Как можно увидеть, последовательность будет отсортирована после $n-1$ итерации.

Параллельная схема

Метод чет-нечетной перестановки:

Алгоритм пузырьковой сортировки в прямом виде достаточно сложен для распараллеливания – сравнение пар значений упорядочиваемого набора данных происходит строго последовательно. В связи с этим для параллельного применения обычно используется не сам этот алгоритм, а его модификация, известная в литературе как метод чет-нечетной перестановки (odd-even transposition). Суть модификации состоит в том, что в алгоритм сортировки вводятся два разных правила выполнения итераций метода – в зависимости от четности или нечетности номера итерации сортировки для обработки выбираются элементы с четными или не-четными индексами соответственно, сравнение выделяемых значений всегда осуществляется с их правыми соседними элементами. Таким образом, на всех нечетных итерациях сравниваются пары $(a_1, a_2), (a_3, a_4), \dots, (a_{n-1}, a_n)$ (при четном n), а на четных итерациях обрабатываются элементы $(a_2, a_3), (a_4, a_5), \dots, (a_{n-2}, a_{n-1})$. После n -кратного повторения итераций сортировки исходный набор данных оказывается упорядоченным.

Блочный алгоритм:

Рассмотрим ситуацию, когда количество вычислительных элементов является меньшим числа упорядочиваемых значений ($p < n$). Разделим сортируемый массив на блоки данных размера n/p . На первом этапе параллельного метода пузырьковой сортировки каждый вычислительный

элемент выполняет сортировку одного из блоков данных при помощи какого-либо быстрого алгоритма; эти действия могут быть выполнены параллельно. На следующем этапе выполняется параллельный алгоритм чет-нечетной перестановки над блоками данных:

-на четных итерациях алгоритма вычислительные элементы с четными индексами $2i$ выполняют операцию «Сравнить и разделить» для двух упорядоченных блоков с номерами $2i$ и $2i+1$,

-на нечетных итерациях алгоритма вычислительные элементы с не-четными индексами $2i+1$ выполняют операцию «Сравнить и разделить» для двух упорядоченных блоков с номерами $2i+1$ и $2i+2$.

После выполнения p итераций чет-нечетной перестановки исходный массив оказывается упорядоченным.