Постановка задачи

1) Реализовать последовательный алгоритм быстрой сортировка
2) Реализовать параллельный алгоритм быстрой сортировки с помощью MPI
3) Провести сравнение скорости работы двух алгоритмов

Сортировка является одной из типовых проблем обработки данных и обычно понимается как задача размещения элементов неупорядоченного набора значений

$$S = \{a_1, a_2, \dots, a_n\}$$

в порядке монотонного возрастания или убывания

$$S \sim S' = \{(a'_1, a'_2, \dots, a'_n) : \ a'_1 \leq a'_2 \leq \dots \leq a'_n\}$$

В данной работе реализован алгоритм быстрой сортировки, предложенной Хоаром (Hoare C.A.R.). Данный алгоритм относится к числу эффективных методов упорядочивания данных и широко используется в практических приложениях.

Метод решения

Алгоритм быстрой сортировки основывается на последовательном разделении сортируемого набора данных на блоки меньшего размера таким образом, что между значениями разных блоков обеспечивается отношение упорядоченности (для любой пары блоков все значения одного из этих блоков не превышают значений другого блока).

На первой итерации метода осуществляется деление исходного набора данных на первые две части – для организации такого деления выбирается некоторый ведущий элемент и все значения набора, меньшие ведущего элемента, переносятся в первый формируемый блок, все остальные значения образуют второй блок набора.

На второй итерации сортировки описанные правила применяются рекурсивно для обоих сформированных блоков и т.д.

При надлежащем выборе ведущих элементов после выполнения log2n итераций исходный массив данных оказывается упорядоченным.

Параллельная схема

Обобщенный алгоритм

В обобщенном алгоритме быстрой сортировки (HyperQuickSort algorithm) в дополнение к обычному методу быстрой сортировки предлагается конкретный способ выбора ведущих элементов. Суть предложения состоит в том, что сортировка блоков данных выполняется в самом начале выполнения вычислений. Кроме того, для поддержки упорядоченности в ходе вычислений над блоками данных выполняется операция слияния, а затем деление полученного блока двойного размера согласно ведущему элементу. Как результат, в силу упорядоченности блоков, при выполнении алгоритма быстрой сортировки в качестве ведущего элемента целесообразнее будет выбирать средний элемент какого-либо блока.

Параллельный алгоритм

Параллельное обобщение алгоритма быстрой сортировки наиболее простым способом может быть получено для случая, когда потоки параллельной программы могут быть организованы в виде N-мерного гиперкуба (т.е. количество вычислительных элементов p = 2^N). Пусть, как и ранее, исходный набор данных логически разделен на 2p блоков одинакового размера n/2p. Тогда блоки данных образуют (N+1)-мерный гиперкуб. Возможный способ выполнения первой итерации параллельного метода при таких условиях может состоять в следующем:

1) выбрать каким – либо образом ведущий элемент (например, в качестве ведущего элемента можно взять среднеарифметическое элементов, расположенных на выбранном ведущем блоке);

2) сформировать пары блоков, для которых необходимо выполнить взаимообмен данными на данной итерации алгоритма: пары образуют блоки, для которых битовое представление номеров отличается только в позиции (N+1);

3) для каждой пары блоков определить вычислительный элемент, который будет выполнять необходимые операции;

4) параллельно выполнить операцию «сравнить и разделить» над всеми парами блоков, в результате такого обмена в блоках, для которых в битовом представлении номера бит позиции N+1 равен 0, должны оказаться части блоков со значениями, меньшими ведущего элемента; блоки с номерами, в которых бит N+1 равен 1, должны собрать, соответственно, все значении данных, превышающие значение ведущего элемента.

В результате выполнения такой итерации сортировки исходный набор оказывается разделенным на две части, одна из которых (со значениями меньшими, чем значение ведущего элемента) располагается в блоках данных, в битовом представлении номеров которых бит N+1 равен 0. Таких блоков всего p и, таким образом, исходный (N+1)–мерный гиперкуб также оказывается разделенным на два гиперкуба размерности N. К этим подгиперкубам, в свою очередь, может быть параллельно применена описанная выше процедура. После (N+1)–кратного повторения подобных итераций для завершения сортировки достаточно упорядочить полученные блоки данных, каждый вычислительный элемент упорядочивает 2 блока.