Лабораторная работа №1. Основы MPI

Вариант 6

1. Распараллелить при помощи MPI скалярное умножение векторов длины *L* типа*double* на *N* процессах. Можно считать, что *L*⋮*N*. Натуральное число *L* получить как аргумент командной строки. Исходные вектора сгенерировать, заполнив их случайными числами из множества {−1; 0} для простоты проверки. Исходные векторы и результат вычислений вывести в консоль. (*До 4 баллов.*)
   1. Написать и выполнить программу (*1 балл*). *Каждый процесс должен сделать максимально возможную часть параллельных вычислений.*
   2. Минимизировать количество данных передаваемых каждому процессу (*1 балл*). *Проконтролируйте, что все переданные данные используются.*
   3. Минимизировать число вызовов MPI (*1 балл*). *Всегда помните, что математическую модель можно по-разному реализовать в коде.*
   4. Поддержать случай, если *L* не делится на *N* (*1 балл*). *Возможно, вам помогут команды ScatterV/GatherV.*
2. Распараллелить при помощи MPI вычитание матриц размера *L*×*M* типа *double* на *N* процессах. Можно считать, что *L*⋮*N*. Натуральные числа *L* и *M* задать в коде как константы. Исходные матрицы сгенерировать, заполнив их случайными целыми числами для простоты проверки. Исходные матрицы и результат вычислений вывести в консоль. (*До 3 баллов.*)
   1. Написать и выполнить программу (*0,75 балла*).
   2. Минимизировать количество данных передаваемых каждому процессу (*0,75 балла*).
   3. Минимизировать число вызовов MPI (*0,75 балла*).
   4. Поддержать случай, если *L* не делится на *N* (*0,75 балла*).
3. Распараллелить при помощи MPI умножение двух матриц размеров *K*×*L* и *L*×*M* типа*double* на *N* процессах. Можно считать, что *L*⋮*N*. Натуральные числа *K*, *L* и *M* считать из консоли. Исходные матрицы сгенерировать, заполнив их случайными числами из множества {−1; 0} для простоты проверки. Исходные матрицы и результат вычислений вывести в консоль. (*До 3 баллов.*)
   1. Написать и выполнить программу (*0,5 балла*).
   2. Минимизировать количество данных передаваемых каждому процессу (*0,5 балла*).
   3. Минимизировать число вызовов MPI (*0,5 балла*).
   4. Поддержать случай, если *L* не делится на *N* (*0,5 балла*).
   5. Засечь время выполнения вычислений на больших *K*, *L* и *M* и разных *N*. Построить график зависимости времени выполнения *T* от *N*. Сделайте выводы (*1 балл*).