КОММЕНТАРИЙ К ПРАКТИЧЕСКОЙ РАБОТЕ №4 «Умножение матрицы на матрицу в MPI 2D решетке» ПО КУРСУ ОПП

Цель 4 практической работы состоит в освоении обучающимися концепции MPI-коммуникаторов и декартовых топологий, а также концепции производных типов данных.

Уточнения:

1) «Компьютеры», о которых идет речь в pdf-документе «Лабораторная работа №3 «Умножение матрицы на матрицу в MPI 2D решетка» описания работы на сайте, моделируются MPI-процессами (каждый MPI-процесс представляет собой 1 «компьютер»).
2) Для простоты можно сгенерировать размеры матриц A[n1 x n2] и B[n2 x n3] так, что n1 кратно p1 *(например, n1=1000*p1)*, а n3 кратно p2.

Общий алгоритм:

1.Создание решетки процессов p1 x p2.

2. Генерация матриц A[n1 x n2] и B[n2 x n3] на процессе с координатами (0;0) как одномерных массивов.

3. Раздача матрицы A по горизонтальным полосам на вертикальную линейку процессов (0;0), (1;0), (2;0), …, (p1 - 1; 0) при помощи MPI_Scatter.

4. Определение нового производного типа данных для выбора из матрицы B вертикальных полос.

5. Раздача матрицы B по вертикальным полосам на горизонтальную линейку процессов (0;0), (0;1), (0;2), …, (0; p2 – 1) таким образом, что каждому процессу высылается только 1 элемент производного типа.
ВНИМАНИЕ: в лабораторной работе указано (рис. 2), что матрица должна раздаваться при помощи MPI_Scatter. Я с вас снимаю данное требование и разрешаю использовать раздачу при помощи коммуникаций типа точка-точка (send-recv). В этом случае, естественно, высылать самому себе процесс (0;0) не должен.

6. Каждый из процессов в левой вертикальной колонке ((1;0), (2;0), …, (p1 - 1; 0)) при помощи MPI_Bcast раздает свою полосу матрицы A всем процессам своей горизонтали. Т.е. процесс (1;0) раздает свою полосу процессам (1;1), (1;2),…

7. То же с полосами матрицы B, которые процессы первой горизонтали раздают по своим вертикальным столбцам решетки процессов (MPI_Bcast).

8. Теперь на каждом процессе есть по полосе A и по столбцу B, перемножаем, получаем миноры C.

9. Собираем всю C на процессе (0;0). В методе реализации этого шага оставляю вам свободу (можно send-recv).

MPI-функции, которые могут пригодиться:

1) **Топологии процессов:**
   a. MPI_Cart_create
   b. MPI_Cart_coords
   c. MPI_Cart_sub
   и не только...
2) **Производные типы данных:**
   a. MPI_Type_commit
   b. MPI_Type_free
   c. MPI_Type_vector
   d. MPI_Type_contiguous
   e. MPI_Type_create_subarray
   f. MPI_Type_create_darray
   g. MPI_Type_struct
   h. MPI_Type_create_resized
   и не только...

Требования:

1) ПИСАТЬ ПРОГРАММУ САМОСТОЯТЕЛЬНО! *(стандартное требование)*
2) Использование декартовой топологии процессов и создание новых коммуникаторов.
3) Использование производных типов данных.
4) Программа должна работать для матрицы B любого содержания - предполагать, что матрица B будет симметричной нельзя.
5) Транспонировать матрицу B нельзя.
6) При раздаче матрицы B по полосам каждый из процессов с координатами (0; x) должен получить **по одному элементу** производного типа, а процесс (0;0), соответственно, высылает процессам по одному элементу производного типа (не обязательно того же самого).

ЗАДАНИЕ К ПРАКТИЧЕСКОЙ РАБОТЕ №3 «Умножение матрицы на матрицу в MPI 2D решетке» ПО КУРСУ ОПП

1. Реализовать параллельный алгоритм умножения матрицы на матрицу при 2D решетке процессов с соблюдением требований (см.выше).

2. Исследовать производительность параллельной программы при фиксированном размере матрицы в зависимости от и размера решетки: 2x12, 3x8, 4x6, 6x4, 8x3, 12x2. Размер матриц подобрать таким образом, чтобы худшее из времен данного набора было не менее 30 сек.

3. Выполнить профилирование программы при использовании 8-и ядер с решетками 2x4, 4x2.