

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М.В.Ломоносова  
Факультет вычислительной математики и кибернетики

## «Распределенные системы»

### ОТЧЕТ

о выполненном задании

студента 427 учебной группы факультета ВМК МГУ

Казанцевой Варвары Денисовны

гор. Москва  
2023 год

## Постановка задачи

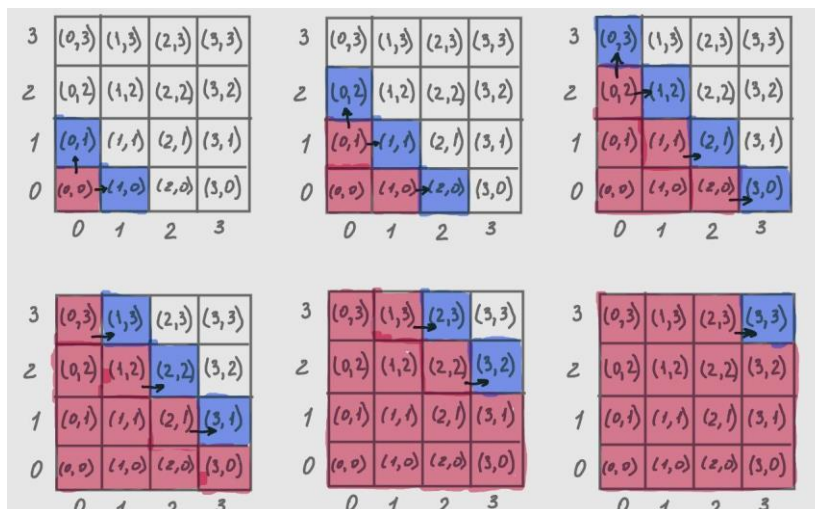
В транспьютерной матрице размером  $4 \times 4$ , в каждом узле которой находится один процесс, необходимо выполнить операцию передачи сообщения длиной  $N$  байт всем процессам от одного (MPI\_BCAST) - процесса с координатами  $(0,0)$ . Реализовать программу, моделирующую выполнение операции MPI\_BCAST на транспьютерной матрице при помощи пересылок MPI типа точка-точка. Получить временную оценку работы алгоритма. Оценить сколько времени потребуется для выполнения операции MPI\_BCAST, если все процессы выдали ее одновременно. Время старта равно 100, время передачи байта равно 1 ( $T_s=100, T_b=1$ ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

## Решение

Операция MPI\_BCAST передает данные от одного участника группы всем членам группы. Ее прототип выглядит следующим образом:

```
int MPI_Bcast(  
    _Inout_ void *buffer,  
    _In_ int count,  
    _In_ MPI_Datatype datatype,  
    _In_ int root,  
    _In_ MPI_Comm comm  
);
```

Каждый процесс (кроме тех, которые расположены в 0-ом столбце) по одному разу отправляет порцию, содержащую необходимые данные для всех процессоров, другому процессу, таким образом охватывая все элементы нашей транспьютерной матрицы. Также условимся, что все процессы могут пересылать данные процессам, которые находятся правее их в топологии, а процессы, расположенные в 0-ом столбце, в добавок могут пересылать еще и вышестоящим (при этом будем отдавать приоритет в отправке тем процессам, которые стоят выше и потом уже отправлять тем, кто расположен правее. Если придерживаться данной стратегии, то мы охватим все 15 процессов за 6 шагов. Визуально это будет выглядеть следующим образом:



Топологию создаем с помощью функции `MPI_Cart_create`, получаем координаты используя `MPI_Cart_coords`. Для отправки и принятия сообщений были использованы функции `MPI_Isend` и `MPI_Recv`.

Теперь оценим время работы алгоритма. Если время старта равно 100, время передачи одного байта – 1 ( $T_s=100$ ,  $T_b=1$ ), то время выполнения по сути сводится к тому, чтобы определить, сколько будут пересылаться данные самому дальнему процессу, который находится в ячейке (3; 3). Этого можно достичь за 6 шагов, и тогда время, чтобы передать данные всем процессам будет вычисляться так:  $time = T_s + n * T_b * 6$ , где  $n$  – размер передаваемого сообщения в байтах.

## Компиляция и запуск

Чтобы скомпилировать программу:

```
mpicc task1_bcast.c -o task1_bcast
```

Чтобы запустить программу:

```
mpiexec -np 16 --oversubscribe ./task1_bcast
```