

Московский Государственный Университет имени М. В. Ломоносова  
Факультет вычислительной математики и кибернетики

**Практикум по курсу  
"Распределенные системы"**

**Релизация MPI\_BCAST при помощи пересылок MPI типа  
точка-точка**

**Отчет**

студента 428 учебной группы

Ковтуна Данилы Петрович

Москва, 2021

## **Содержание**

- 1) Постановка задачи
- 2) Описание алгоритма
- 3) Реализация алгоритма
- 4) Запуск и проверка программы
- 5) Временная оценка

## **Постановка задачи**

В транспьютерной матрице размером  $8 \times 8$ , 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.

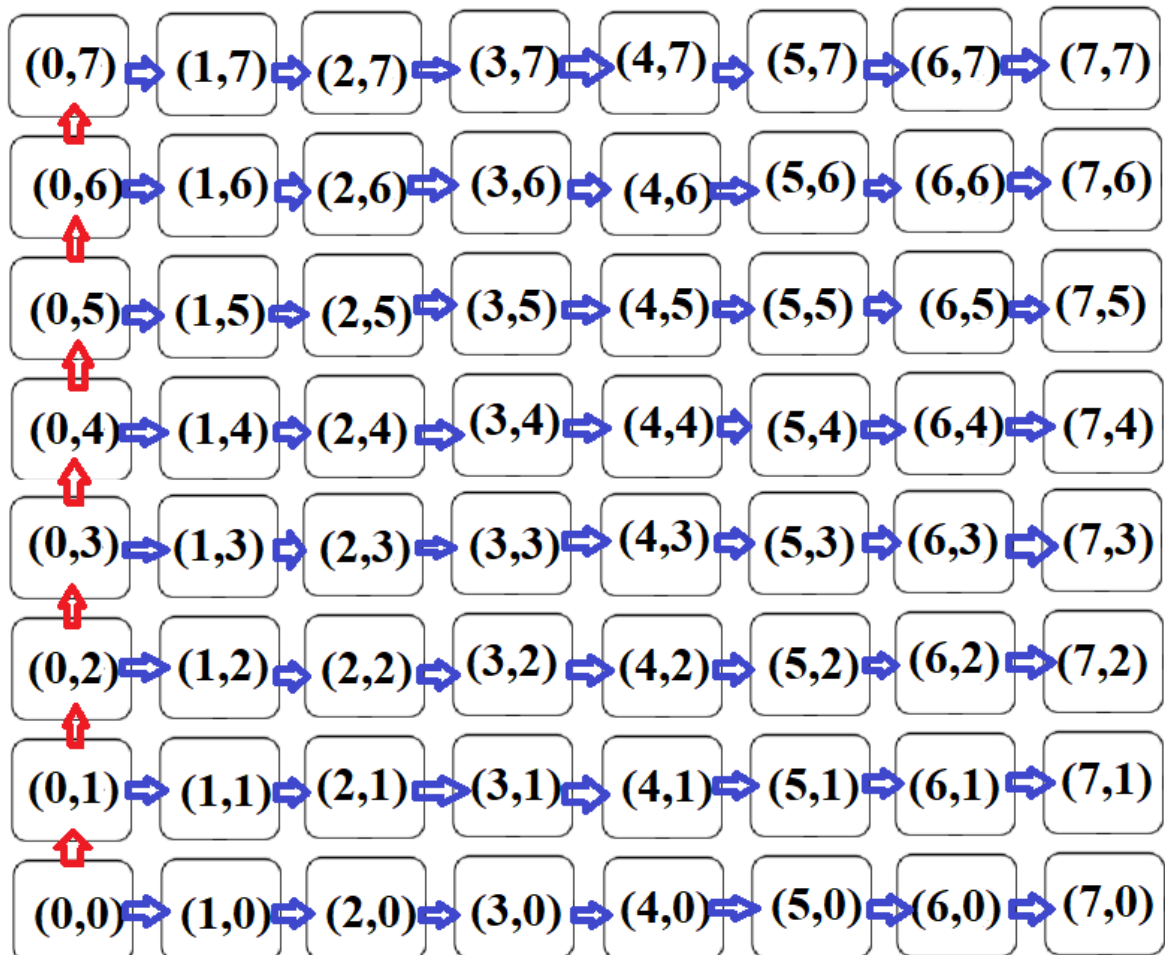
Получить временную оценку работы алгоритма. Оценить сколько времени потребуется для выполнения операции MPI\_BCAST, если все процессы выдали ее одновременно. Время старта равно 100, время передачи байта равно 1 ( $T_s=100, T_b=1$ ). Процессорные операции, включая чтение из памяти и запись в память, считаются бесконечно быстрыми.

## Описание алгоритма

MPI\_Bcast - рассылка информации от одного процесса всем остальным членам некоторой области.

Распространять сообщение будем по следующей стратегии:

- сначала отправим его вверх по прямой  $x = 0$ , потом по прямым  $y = \{0, 1, 2, 3, 4, 5, 6, 7\}$  вправо.
- При этом процессам, находящимся на прямой  $x = 0$ , не нужно ждать доставки сообщения
- до  $(0, 7)$  — они будут отправлять сообщения вправо сразу после того, как отправят вверх.



## Реализация алгоритма

```
#include <mpi.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

static const char* VERY_LONG_MESSAGE = "abc";

static const int REORDER = 0;
static const int NDIMS = 2;
static const int DIMS[2] = {8, 8};
static const int PERIODS[2] = {0, 0};

int main(int argc, char *argv[]) {
    MPI_Init(&argc, &argv);
    MPI_Comm comm_cart;
    int size;
    int rank;
    int coords[2];
    int length = strlen(VERY_LONG_MESSAGE) + 1;
    char *m = (char *) malloc(length * sizeof(char));

    MPI_Cart_create(MPI_COMM_WORLD, NDIMS, DIMS, PERIODS, REORDER, &comm_cart);
    MPI_Comm_size(comm_cart, &size);
    MPI_Comm_rank(comm_cart, &rank);
    MPI_Cart_coords(comm_cart, rank, NDIMS, coords);

    if (rank == 0) {
        strcpy(m, VERY_LONG_MESSAGE);
    }

    if (coords[0] == 0) {
        if (coords[1] != 7) {
            // Отправка сообщения "наверх"
            int dest;
            int dest_coords[2] = {0, coords[1] + 1};
            MPI_Cart_rank(comm_cart, dest_coords, &dest);
            MPI_Send(m, length, MPI_CHAR, dest, length - 1, comm_cart);
        }
        if (coords[1] != 0) {
            // Получение сообщения "снизу"
            MPI_Status status;
            int source_coords[2] = {0, coords[1] - 1};
            int source;
            MPI_Cart_rank(comm_cart, source_coords, &source);
            MPI_Recv(m, length, MPI_CHAR, source, length - 1, comm_cart,
&status);
        }
    } else {
```

```
    // Получение сообщения "слева"
    MPI_Status status;
    int source_coords[2] = {coords[0] - 1, coords[1]};
    int source;
    MPI_Cart_rank(comm_cart, source_coords, &source);
    MPI_Recv(m, length, MPI_CHAR, source, length - 1, comm_cart, &status);
}

if (coords[0] != 7) {
    // Отправка сообщения "направо"
    int dest;
    int dest_coords[2] = {coords[0] + 1, coords[1]};
    MPI_Cart_rank(comm_cart, dest_coords, &dest);
    MPI_Send(m, length, MPI_CHAR, dest, length - 1, comm_cart);
}

printf("%d: %s\n", rank, m); fflush(stdout);

MPI_Finalize();
return 0;
}
```

## Запуск и проверка программы

```
danila@danila-Lenovo-ideapad-330-15IKB:~/4-course/distributed-systems$ mpicc my_MPI_BCAST.c -o my_MPI_BCAST
danila@danila-Lenovo-ideapad-330-15IKB:~/4-course/distributed-systems$ mpiexec -np 64 --oversubscribe ./my_MPI_BCAST | sort
-----
WARNING: A deprecated command line option was used.

  Deprecated option:  --oversubscribe
  Corrected option:   --map-by :oversubscribe:OVERSUBSCRIBE

We have updated this for you and will proceed. However, this will be treated
as an error in a future release. Please update your command line.
-----

***** Corrected cmd line: prterun --map-by :oversubscribe:OVERSUBSCRIBE --mca btl tcp,self --np 64 ./my_MPI_BCAST
```

```
0: abc
10: abc
11: abc
12: abc
13: abc
14: abc
15: abc
16: abc
17: abc
18: abc
19: abc
1: abc
20: abc
21: abc
22: abc
23: abc
24: abc
25: abc
26: abc
27: abc
28: abc
29: abc
2: abc
30: abc
31: abc
32: abc
33: abc
34: abc
35: abc
36: abc
37: abc
38: abc
39: abc
3: abc
40: abc
41: abc
42: abc
43: abc
44: abc
45: abc
46: abc
47: abc
48: abc
49: abc
4: abc
```

```
50: abc
51: abc
52: abc
53: abc
54: abc
55: abc
56: abc
57: abc
58: abc
59: abc
5: abc
60: abc
61: abc
62: abc
63: abc
6: abc
7: abc
8: abc
9: abc
```

```
danila@danila-Lenovo-ideapad-330-15IKB:~/4-course/distributed-systems$
```

Как мы видим, все процессы получили корректное сообщение и вывели его в stdout его.

## Временная оценка

Дольше всего сообщение будет идти по маршруту:

$$(0,0) \rightarrow (0,1) \rightarrow \dots \rightarrow (0,7) \rightarrow (1,7) \rightarrow \dots (7,7)$$

На его доставку потребуется 14 пересылок

От одного узла до другого будет затрачено

$$(T_s + N \cdot T_b)$$

Значит, все процессы получают сообщение через

$$14 * (T_s + N \cdot T_b) = 14 * (100 + N)$$