

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «ПА»
Тема: Коллективные операции

Студент гр. 8304

Сергеев А.Д.

Преподаватель

Татаринов Ю.С.

Санкт-Петербург

2020

Задание.

В каждом процессе дан набор из $K + 5$ чисел, где K — количество процессов. Используя функцию `MPI_Reduce` для операции `MPI_MIN`, найти минимальное значение среди элементов данных наборов с одним и тем же порядковым номером и вывести полученные минимумы в главном процессе.

Описание алгоритма.

В каждом процессе был сформирован массив из $k+5$ целых чисел. После этого с помощью функции `MPI_Reduce` были найдены минимумы среди 1-ых, 2-ых и т.д. элементов, массив полученных структур помещается в `answers`. В нулевом процессе выводятся значения минимумов.

Исходный код программы.

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

#define array_upper_limit 1000000.0

void populate_integer_array(int* arr, int size, int rank) {
    srand(time(NULL) + rank);
    for (int i = 0; i < size; i++) arr[i] = (int) ((array_upper_limit /
RAND_MAX) * rand());
}

int main(int argc, char* argv[]) {
    int ProcNum, ProcRank;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &ProcNum);
    MPI_Comm_rank(MPI_COMM_WORLD, &ProcRank);
    int k = ProcNum + 5;
    double start_time = MPI_Wtime();

    int* containment = malloc(k * sizeof(int));
    populate_integer_array(containment, k, ProcRank);

    int* answers = malloc(k * sizeof(int));
    MPI_Reduce(containment, answers, k, MPI_INT, MPI_MIN, 0, MPI_COMM_WORLD);

    if (ProcRank == 0) {
        for (int i = 0; i < k; ++i)
```

```

        printf("Root process: minimum for index %d is %d\n", i, answers[i]);
        printf("It took %lf sec\n", MPI_Wtime() - start_time);
    }

    free(containment);
    free(answers);
    MPI_Finalize();
    return 0;
}

```

Результаты работы программы.

```

Root process: minimum for index 0 is 125146
Root process: minimum for index 1 is 58747
Root process: minimum for index 2 is 601959
Root process: minimum for index 3 is 276348
Root process: minimum for index 4 is 197193
Root process: minimum for index 5 is 71507
Root process: minimum for index 6 is 152683
Root process: minimum for index 7 is 173509
It took 0.000745 sec

```

Рисунок 1 — Результат работы программы для 3 процессов

```

Root process: minimum for index 0 is 240399
Root process: minimum for index 1 is 83931
Root process: minimum for index 2 is 107352
Root process: minimum for index 3 is 81908
Root process: minimum for index 4 is 61216
Root process: minimum for index 5 is 67172
Root process: minimum for index 6 is 20424
Root process: minimum for index 7 is 60328
Root process: minimum for index 8 is 36612
Root process: minimum for index 9 is 147432
Root process: minimum for index 10 is 103856
Root process: minimum for index 11 is 143218
It took 0.014015 sec

```

Рисунок 2 — Результат работы программы для 7 процессов

```
Root process: minimum for index 0 is 47928
Root process: minimum for index 1 is 44111
Root process: minimum for index 2 is 69173
Root process: minimum for index 3 is 10933
Root process: minimum for index 4 is 138273
Root process: minimum for index 5 is 204945
Root process: minimum for index 6 is 65154
Root process: minimum for index 7 is 28396
Root process: minimum for index 8 is 11538
Root process: minimum for index 9 is 81130
Root process: minimum for index 10 is 151420
Root process: minimum for index 11 is 10842
Root process: minimum for index 12 is 34021
Root process: minimum for index 13 is 11951
Root process: minimum for index 14 is 91572
Root process: minimum for index 15 is 71817
Root process: minimum for index 16 is 61288
Root process: minimum for index 17 is 60579
Root process: minimum for index 18 is 13370
Root process: minimum for index 19 is 205271
Root process: minimum for index 20 is 48246
Root process: minimum for index 21 is 137388
Root process: minimum for index 22 is 7250
Root process: minimum for index 23 is 130493
It took 0.065626 sec
```

Рисунок 3 — Результат работы программы для 19 процессов

```
Root process: minimum for index 57 is 3433
Root process: minimum for index 58 is 3701
Root process: minimum for index 59 is 112
Root process: minimum for index 60 is 1906
Root process: minimum for index 61 is 31995
It took 0.107511 sec
```

Рисунок 4 — Результат работы программы для 57 процессов (некоторые строки опущены)

```
Root process: minimum for index 101 is 12849
Root process: minimum for index 102 is 207
Root process: minimum for index 103 is 6507
Root process: minimum for index 104 is 2040
Root process: minimum for index 105 is 14867
It took 0.414259 sec
```

Рисунок 5 — Результат работы программы для 101 процесса (некоторые строки опущены)

График зависимости времени выполнения программы от числа процессов.

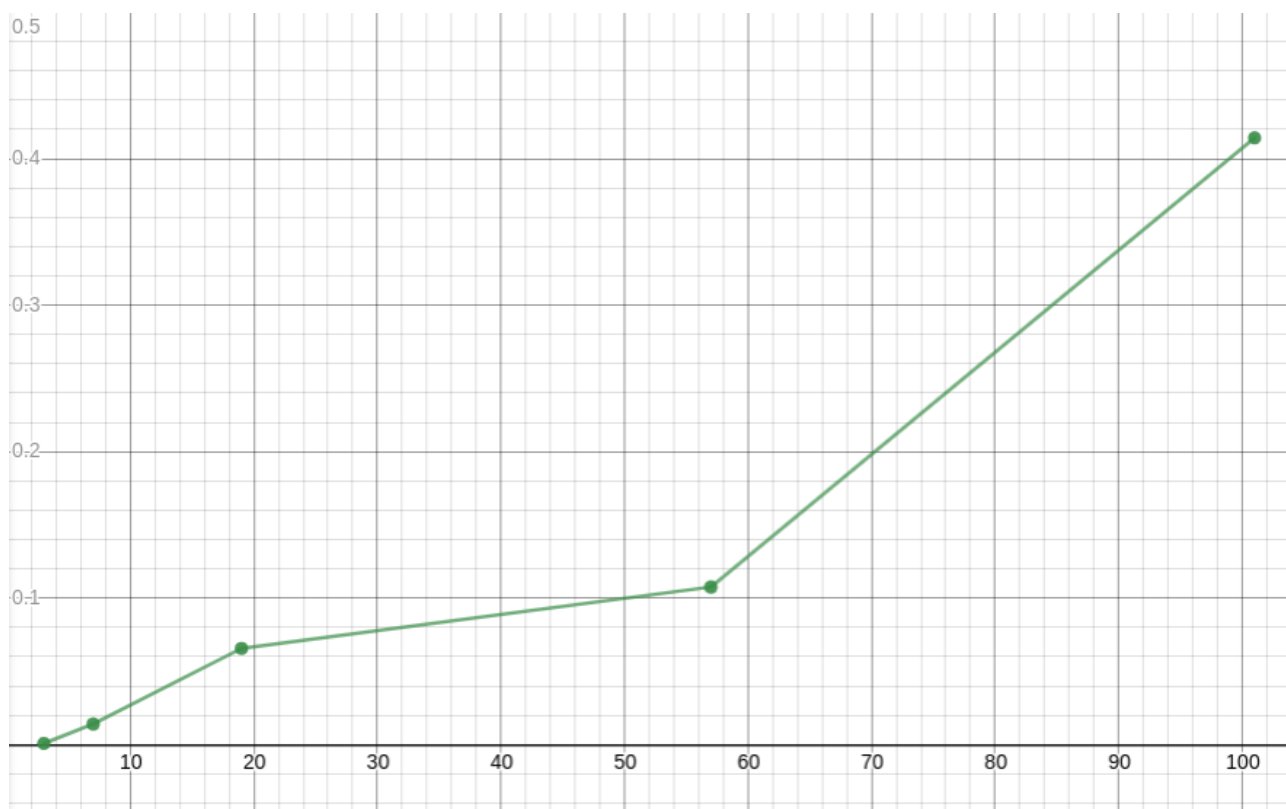


Рисунок 6 — График зависимости времени выполнения программы от числа процессов

Выводы.

Создана программа, поиска минимума среди элементов массивов, хранящихся в разных процессах, с один индексом. Изучены принципы работы функции редукции, проводящей операции над переменными разных процессов.