**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №5**

**по дисциплине «Системы параллельной обработки данных»**

**Тема: «Виртуальные топологии»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9303 |  | Королёв С.Ю. |
| Преподаватель |  | Татаринов Ю.С. |

Санкт-Петербург

2023

**Цель работы**

Базовое освоение и изучение работы с виртуальными топологиями в рамках библиотеки MPI.

**Формулировка задания**

Количество процессов K равно 8 или 12, в каждом процессе дано вещественное число. Определить для всех процессов декартову топологию в виде трехмерной решетки размера 2 × 2 × K/4 (порядок нумерации процессов оставить прежним). Интерпретируя эту решетку как две матрицы размера 2 × K/4 (в одну матрицу входят процессы с одинаковой первой координатой), расщепить каждую матрицу процессов на K/4 одномерных столбцов. Используя одну коллективную операцию редукции, для каждого столбца процессов найти произведение исходных чисел и вывести найденные произведения в главных процессах каждого столбца.

**Ход работы**

Для изучения реализации виртуальных топологий в библиотеке MPI был создан скрипт на языке программирования С++ для запуска и обработки результатов запусков процессов.

**Листинг итоговой программы**

#include <iostream>

#include "mpi.h"

int main(int argc, char\* argv[]) {

MPI\_Init(&argc, &argv);

int total\_procs, proc\_rank;

MPI\_Comm\_size(MPI\_COMM\_WORLD, &total\_procs);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &proc\_rank);

srand(time(NULL) + proc\_rank); // генератор случайных чисел для каждого процесса с использованием его ранга

int secret\_number = rand() % 9 + 1; // случайное число от 1 до 9 для каждого процесса

int grid\_dim\_1 = 2, grid\_dim\_2 = 2, grid\_dim\_3 = total\_procs / 4; // размеры трехмерной решетки процессов

int grid\_dims[3] = { grid\_dim\_1, grid\_dim\_2, grid\_dim\_3 }; // массив с этими размерами

int grid\_periods[3] = { 0, 0, 0 }; // будут ли циклические периодические условия в каждом измерении

// коммуникатор для трехмерной решетки

MPI\_Comm three\_dim\_grid\_comm;

MPI\_Cart\_create(MPI\_COMM\_WORLD, 3, grid\_dims, grid\_periods, 0, &three\_dim\_grid\_comm);

// координаты текущего процесса в трехмерной решетке

int grid\_coords[3];

MPI\_Cart\_coords(three\_dim\_grid\_comm, proc\_rank, 3, grid\_coords);

printf("Process %2d. 3D coordinates: (%d %d %d). Secret number: %d\n", proc\_rank, grid\_coords[0], grid\_coords[1], grid\_coords[2], secret\_number);

// коммуникатор для столбца в трехмерной решетке

int column\_subdims[3] = { 0, 0, 1 };

MPI\_Comm column\_comm;

MPI\_Cart\_sub(three\_dim\_grid\_comm, column\_subdims, &column\_comm);

// ранг текущего процесса в новом коммуникаторе

int column\_rank;

MPI\_Comm\_rank(column\_comm, &column\_rank);

printf("Old rank: %d, New rank in column communicator: %d\n", proc\_rank, column\_rank);

// умножение всех сгенерированных чисел в каждом столбце трехмерной решетки

int in[1] = { secret\_number }, out[1];

MPI\_Reduce(in, out, 1, MPI\_INT, MPI\_PROD, 0, column\_comm);

// результат умножения только для процесса с рангом 0 в новом коммуникаторе

if (column\_rank == 0) {

printf("\*\*\* Process %2d result after multiplication in the column: %d \*\*\*\n", proc\_rank, out[0]);

}

MPI\_Finalize();

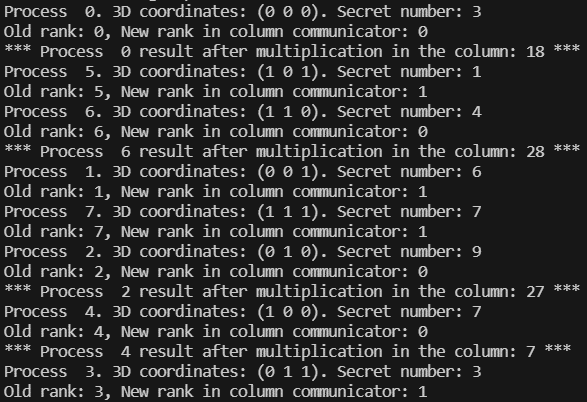
return 0;

}

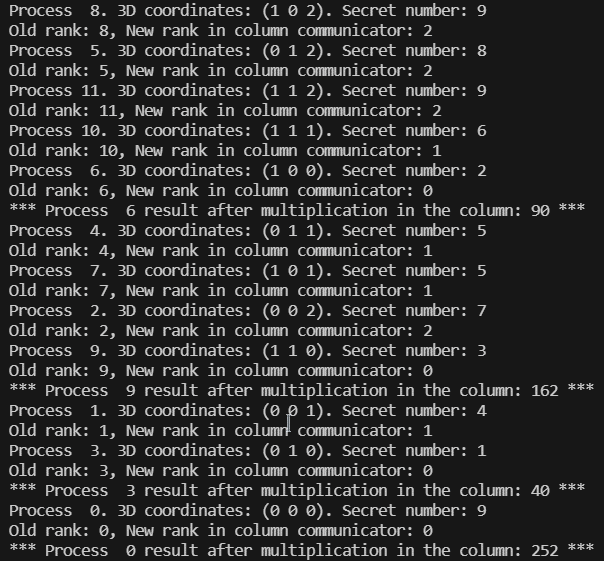
**Результаты работы исходной программы для разного числа процессов**

Главный процесс в каждом из столбцов топологии выводит результат на экран. Главным считается процесс с рангом 0. По ходу выполнения программы выводятся значения главных параметров: координаты процессов в первичной топологии вместе с рангами и сгенерированным числами, отображения старого ранга на новый в столбцах, результат коллективной операции редукции и ранг процесса в коммуникаторе COMM\_WORLD.

Для 8 процессов:



Для 12 процессов:



**Краткое описание выбранного алгоритма решения**

Для построения начальной пространственной топологии 2х2хK/4 (для 8 потоков, например, будет 2х2х2) используется функция MPI\_Cart\_create. Затем созданную матрицу расщепили на столбцы по 2 элемента (для 8 потоков) с помощью функции MPI\_Cart\_sub.

Для проверки правильности разбиения с помощью сравнения полученных координат процессов в декартовой топологии с теоретическими использовалась функция MPI\_Cart\_coords.

В каждом процессе было сгенерировано случайное число от 1 до 9, недоступное остальным процессам. Для окончательного решения поставленной задачи по нахождению произведения чисел в полученных столбцах топологии была использована функция MPI\_Reduce.

**Формальное описание выбранного алгоритма решения с использованием аппарата сетей Петри**

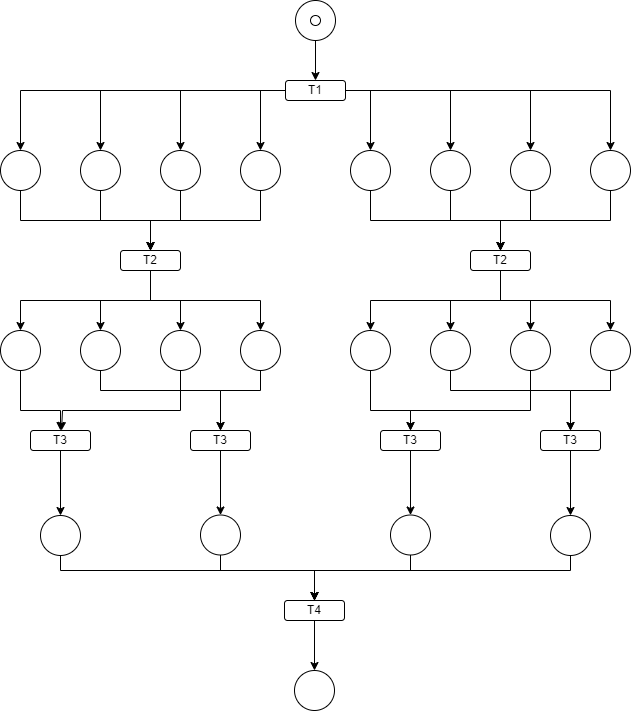
Условные обозначения:

T1 – разбиение на процессы

T2 – разделение на 2 части

T3 – редукция по столбцам

T4 – завершение



**График эффективности**

Поскольку количество процессов в задаче задано изначально (8 или 12) и является константным, график зависимости времени работы программы для разного количества процессов не имеет смысла.

**Вывод**

В ходе выполнения данной лабораторной работы была на базовом уровне освоена и изучена работа с виртуальными топологиями в рамках библиотеки MPI. Были применены методы создания виртуальной топологии, на основании преобразованной топологии были произведены вычисления с помощью операции редукции. Была успешно реализована требуемая функциональность программы.