МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационные системы

Бариев Эмин Юсуфович

Институт информационных технологий и управления в технических системах

курс 4 группа ИС/б-16-2

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №2

По дисциплине: «ТРСиПВ»

По теме: «Исследование коллективного типа передачи данных, групп и коммуникаторов в MPI»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

ст.преподаватель Дрозин А. Ю.

(должность) (подпись) (инициалы,фамилия)

Севастополь 2019

1 ЦЕЛЬ РАБОТЫ

Исследовать способы обмена данными между процессами в режиме широковещания или группового обмена с использованием MPI-функций.

2 ПОСТАНОВКА ЗАДАЧИ

Вариант – 1

Реализовать блочный алгоритм распределенного параллельного перемножения матриц и с размерами (8\*5) и (5\*3) соответственно. Вид распределяемых между процессами блоков представлен на рисунке 1.



Рисунок 1– Перемножение матриц

3 ХОД РАБОТЫ

3.1 Код программы на языке С++

#include <mpi.h>

#include <stdio.h>

#include <iostream>

#include <string>

**using** **namespace** std;

**void** MultiplicationMatrix(**int** matrA2[2][5],**int** matrB[5][3], **int** semiResult[2][3]);

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

**int** rank;

MPI\_Status status;

**int** arrayA[8][5]={

{ 2, 2, 2, 2, 2 },

{ 1, 1, 1, 1, 1 },

{ 3, 3, 3, 3, 3 },

{ 4, 4, 4, 4, 4 },

{ 5, 5, 5, 5, 5 },

{ 6, 6, 6, 6, 6 },

{ 2, 2, 2, 2, 2 },

{ 0, 0, 0, 0, 0 },

};

**int** arrayB[5][3]={

{ 3, 3, 3 },

{ 3, 3, 3 },

{ 3, 3, 3 },

{ 3, 3, 3 },

{ 3, 3, 3 },

};

**int** arrayA2[2][5];

**int** arrayC[8][3];

**int** resultArray[2][3];

MPI\_Init(&argc, &argv);

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

MPI\_Scatter(&arrayA, 10, MPI\_INT, &arrayA2, 10, MPI\_INT, rank, MPI\_COMM\_WORLD);

MPI\_Bcast(&arrayB, 15, MPI\_INT, 0, MPI\_COMM\_WORLD);

MultiplicationMatrix(arrayA2, arrayB, resultArray);

MPI\_Gather(resultArray, 6, MPI\_INT, &arrayC, 6, MPI\_INT, 0, MPI\_COMM\_WORLD);

MPI\_Barrier(MPI\_COMM\_WORLD);

**if**(rank == 0){

cout<<"------Результатирующая матрица:------"<<endl;

**for** (**int** i = 0; i < 8; i++){

**for**(**int** j = 0; j < 3 ; j++){

cout<<arrayC[i][j]<<" ";

}

cout<<endl;

}

}

MPI\_Finalize();

**return** 0;

}

**void** MultiplicationMatrix(**int** arrayA2[2][5],**int** arrayB[5][3], **int** semiResult[2][3]){

**for**(**int** k=0; k<2; k++){

**for** (**int** i = 0; i < 3; i++){

**for**(**int** j = 0; j < 5 ; j++){

semiResult[k][i] += arrayA2[k][j] \* arrayB[j][i];

}

}

}

}

3.2 Результаты выполнения программы

**------Результатирующая матрица:------**

**30 30 30**

**15 15 15**

**45 45 45**

**60 60 60**

**75 75 75**

**90 90 90**

**30 30 30**

**0 0 0**

**Program ended with exit code: 0**

ВЫВОДЫ

В ходе выполнения лабораторной работы были исследованы способы обмена данными между процессами в режиме широковещания или группового обмена с использованием MPI-функций.

Была реализована программа, реализующая блочный алгоритм распределенного параллельного перемножения матриц и с размерами (8\*5) и (5\*3) соответственно.