FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER EDUCATION

ITMO UNIVERSITY

Report

on the practical task No. 9, 10, 11

Performed by

*Alexander Yamoldin*

*J4134c*

St. Petersburg 2021

# Goal

Understand basic MPI C++ syntax, use MPI library to realize the vector addition operation, realize the receiving and sending operations without blocking and realizing procedure of combined reception and transmission of messages .

# Formulation of the problem

In the Assignment 9 need to Write an MPI program in which the global vector addition operation is modeled by a doubling (cascade) scheme using point-to-point data transfers. Compare the execution time of such a simulation using the MPI\_Reduce procedure on as many processes as possible. Each process stores an array of 1,000,000 elements equal to ‘1’.

In the Assignment 10 need to complete the program Assignment10.c. Compile and run it.

In the Assignment 11 need to based on Assignment 10, write a program for ring topology exchange using the MPI\_Sendrecv() function.

In situations where you need to exchange data between processes, it is safer to use the overlaid MPI\_Sendrecv operation. The MPI\_Sendrecv function combines the execution of the send and receive operations. Both operations use the same communicator, but message IDs may differ. The location of the received and transmitted data in the address space of the process should not overlap. The data sent can be of different types and lengths.

In cases when it is necessary to exchange data of the same type with replacement of the sent data with the received ones, it is more convenient to use the MPI\_Sendrecv\_replace function. In this operation, the data sent from the buf array is replaced with the received data.

The special address MPI\_PROC\_NULL can be used for source and dest in data transfer operations. Communication operations with such an address do nothing. The use of this address is convenient instead of using logical constructs to analyze the conditions to send / read a message or not.

# Results

The code of the Assignment 9 can be found in

<https://github.com/AAYamoldin/TrainingPrograms/blob/master/institute_c_programs/ITMO_Parallel_Algorithm/Task_9/Assignment9.cpp>

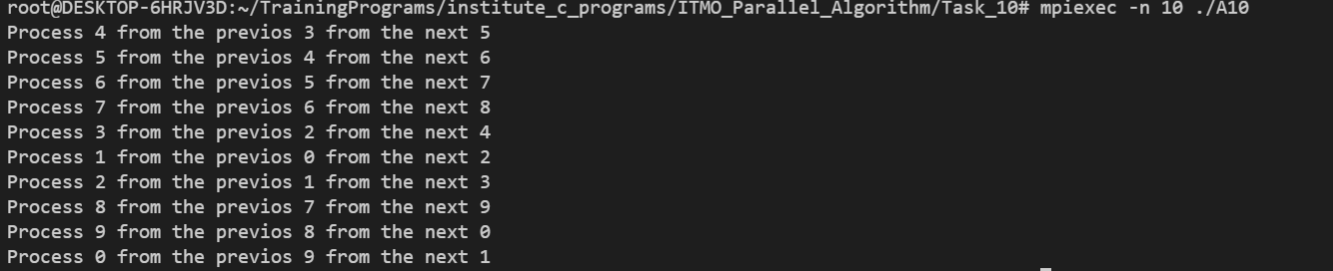
The result of the program in the table below:

|  |  |  |
| --- | --- | --- |
| Num process | Cascade time, s | MPI\_Reduce time, s |
| 2 | 0.013510 | 0.006828 |
| 4 | 0.004889 | 0.006164 |
| 8 | 0.007895 | 0.005961 |
| 10 | 0.008348 | 0.008476 |

The code of the Assignment 10 can be found in

<https://github.com/AAYamoldin/TrainingPrograms/blob/master/institute_c_programs/ITMO_Parallel_Algorithm/Task_10/Assignment10.c>

The result of the program is the picture below:



The code of the Assignment 11 can be found in

<https://github.com/AAYamoldin/TrainingPrograms/blob/master/institute_c_programs/ITMO_Parallel_Algorithm/Task_11/Assignment11.cpp>

The result in the picture and table below:

