# Лабораторная работа №6

# Обмен сообщениями с использованием технологии MPI

## 2.1 Цель работы

Изучение типов коммутации и способов обмена сообщения технологии МРІ.

**2.2** **Теоретические сведения**

Пример программы с передачей текста от нескольких машин и множественным приемом одной из машин:

import mpi.\*;

import java.io.\*;

import java.nio.ByteBuffer;

public class MPITest

{

 public static void main(String[] args)

 {

 MPI.Init(args);

 int me = MPI.COMM\_WORLD.Rank();

 int tasks = MPI.COMM\_WORLD.Size();

 MPI.COMM\_WORLD.Barrier();

 if(me == 0)

 {

 Cat cat = new Cat("Tom", 15);

 cat.Speak();

 ByteBuffer byteBuff = ByteBuffer.allocateDirect(2000 + MPI.SEND\_OVERHEAD);

 MPI.Buffer\_attach(byteBuff);

 try

 {

 ByteArrayOutputStream bos = new ByteArrayOutputStream();

 ObjectOutput out = null;

 out = new ObjectOutputStream(bos);

 out.writeObject(cat);

 byte[] bytes = bos.toByteArray();

 System.out.println("Serialized to " + bytes.length);

 MPI.COMM\_WORLD.Isend(bytes, 0, bytes.length, MPI.BYTE, 1, 0);

 }

 catch(IOException ex)

 {

 }

 }

 else

 {

 byte[] bytes = new byte[2000];

 Cat recv = null;

 MPI.COMM\_WORLD.Recv(bytes, 0, 2000, MPI.BYTE, MPI.ANY\_SOURCE, 0);

 ByteArrayInputStream bis = new ByteArrayInputStream(bytes);

 ObjectInput in = null;

 try

 {

 in = new ObjectInputStream(bis);

 Object obj = in.readObject();

 recv = (Cat)obj;

 recv.Speak();

 }

 catch(IOException ex)

 {

 }

 catch(ClassNotFoundException cnf)

 {

 }

 }

 MPI.COMM\_WORLD.Barrier();

 MPI.Finalize();

 }

}

Пример программы обмена информацией между несколькими машинами:



Пример 3 программы с коллективной передачей:



Задание к работе:

1. Ознакомиться с документацией методов Send(), Recv(), Scatter(), Gather() и других, обеспечивающих коммутацию между процессами и/или компьютерами.
2. Реализовать примеры программ, приведенные в теоретических сведениях.
3. Составить программу:
	1. программу, реализующую каскадный алгоритм вычисления суммы заданных значений с использованием коммутации «точка-точка»;
	2. программу, реализующую каскадный алгоритм вычисления суммы заданных значений с использованием коллективной коммутации;
	3. программу, реализующую каскадный модифицированный алгоритм вычисления суммы заданных значений с использованием коммутации «точка-точка»;
	4. программу, реализующую каскадный модифицированный алгоритм вычисления суммы заданных значений с использованием коллективной коммутации;
4. Исследовать время выполнения каскадного и модифицированного каскадного алгоритмов в зависимости от количества суммируемых значений для каждого типа коммутации.