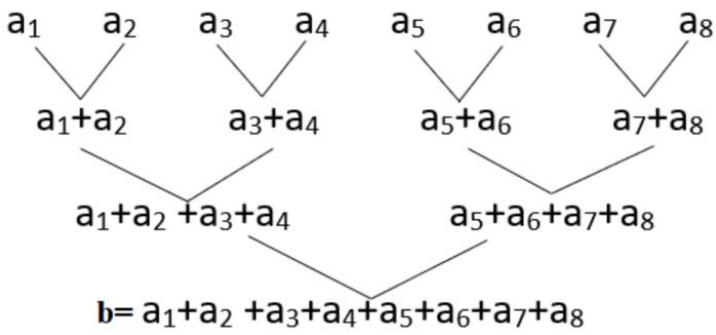
Assignment 9. MPI_Reduce.

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’.

The resulting array is saved, for example, on process 0. To check the result, print the first 5 elements for both methods to the console.

Doubling scheme



MPI_Reduce

The **MPI_Reduce** function concatenates the input buffer entries of each process in a group using the ***op*** operation and returns the concatenated value to the root process's output buffer.

The syntax for the **MPI_Reduce** reduction function is:

int **MPI_Reduce** (void * **sendbuf**, void * **recvbuf**, int **count**, MPI_Datatype **datatype**,

MPI_Op **op**, int **root**, MPI_Comm **comm**)

IN **sendbuf** - address of the beginning of the input buffer;

OUT **recvbuf** - address of the beginning of the result buffer (used only in the receiving process root);

IN **count** - the number of elements in the input buffer;

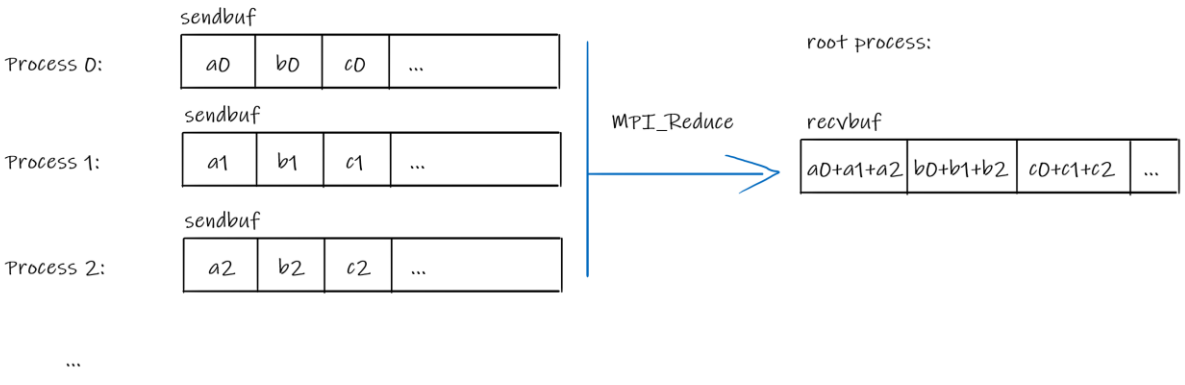
IN **datatype** - the type of elements in the input buffer;

IN **op** - the operation by which the reduction is performed;

IN **root** - number of the receiving process of the operation result;

IN **comm** - communicator.

For example, MPI_Reduce(&a, &b, 3, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);


sendbuf
Process 0:
a0
b0
c0
...
sendbuf
Process 1:
a1
b1
c1
...
sendbuf
Process 2:
a2
b2
c2
...
...
MPI_Reduce
root process:
recvbuf
a0+a1+a2
b0+b1+b2
c0+c1+c2
...