The Ministry of Education and Science of the Russian Federation
Lobachevsky State University of Nizhni Novgorod
Computing Mathematics and Cybernetics faculty

The competitiveness enhancement program of the Lobachevsky State University of Nizhni Novgorod among the world's research and education centers

Strategic initiative “Achieving leading positions in the field of supercomputer technology and high-performance computing”

Parallel Programming
for Multiprocessor Distributed Memory Systems

01 Lecture
The Fundamentals of MPI

Brief description

Nizhni Novgorod
2014

01_LECTURE. THE FUNDAMENTALS OF MPI

OBJECTIVES

An objective of the lecture is to study the fundamental principles of the MPI technology and the basic set of functions sufficient for developing simple MPI programs.

ABSTRACT

The basic concepts of MPI technology are discussed. Necessary definitions are given. The fundamentals of MPI are considered including MPI program initialization, data transmission operation. An example of simple MPI program is introduced as well as an introduction into collective operations.

BRIEF OVERVIEW

It is mentioned at the very beginning of the lecture that MPI is a message passing interface. It is currently one of the basic approaches to develop parallel programs for the distributed memory computer systems. The use of MPI makes possible to distribute the computational load and arrange the information interaction, data transmission among the processors. The term MPI means on the one hand the standard to which the software of arranging message passing must meet. On the other hand, this term denotes the program libraries, which provide the possibility of message passing and meet all the requirements of the standard.

At the beginning a number of concepts and definitions, which are the basic ones for the standard MPI, are discussed. A parallel program as a number of simultaneously executed processes is presented. These processes may be carried out on different processors but several processes may be located on a processor (in these cases they are executed in the time-sharing mode). A brief characteristic of the concept needed for the description of the message passing operations is given further. The terms of data types, the communicators and virtual topologies are introduced.

After that, a brief and simple introduction into the development of MPI based parallel programs is given. Necessary minimum number of MPI functions is introduced: initialization, finalization, MPI_COMM_WORLD communicator, obtaining the process rank in and size of the communicator, basic data transmission functions MPI_Send and MPI_Recv. The typical structure of MPI program is described. The material of this section is a good basis for starting the development of parallel programs with various complexity.

Next section presents usage of previously discussed functions in the first MPI program “Hello World”. Each process in this program gets its rank and sends it via MPI_Send function to the process with rank 0. The zero process receives all messages and prints them out. Two possible way of message receiving from a set of processes via MPI_Recv function is discussed:

- In strong order, from the process with rank 1 at first, form the process with rank 2 at second and so on.
- In unknown order when messages are received in order of appearance instead of in order of ranks.

The last section introduces collective communications and discusses the example problem of summation. Data broadcast operation is considered as well as the difference between broadcast usual implementations in MPI libraries and simple implementations using a loop of sending operations from root process. At the end full description of data reduction operation is given.

FOR STUDENTS

There are a number of sources, which provide information about MPI. First of all, this is the internet resource, which describes the standard MPI: http://www.mpiforum.org. One of the most widely used MPI realizations, the library MPICH, is presented on http://www.mpich.org/.

The following works may be recommended: Pacheco (1996), Snir, et al. (1996), Group, et al. (1999a). The description of the standard MPI-2 may be found in Group, et al. (1999b). The description of standard MPI-3 may be found at www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf.

We may also recommend the work by Quinn (2004), which described a number of typical problems of parallel programming for the purpose of studying MPI.

REFERENCES

1. The internet resource, which describes the standard MPI: http://www.mpiforum.org.
2. One of the most widely used MPI realizations, the library MPICH, is presented on http://www.mpich.org.
3. Quinn, M.J. (2004). Parallel Programming in C with MPI and OpenMP. – New York, NY: McGraw-Hill.
4. Pacheco, P. (1996). Parallel Programming with MPI. - Morgan Kaufmann.
5. Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J. (1996). MPI: The Complete Reference. – MIT Press, Boston, 1996.

6. Group, W., Lusk, E., Skjellum, A. (1999a). Using MPI – 2nd Edition: Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation). – MIT Press.
7. Group, W., Lusk, E., Thakur, R. (1999b). Using MPI-2: Advanced Features of the Message Passing Interface (Scientific and Engineering Computation). – MIT Press.

EXERCISES

1. Develop a program for finding the minimum (maximum) value of the vector elements.
2. Develop a program for computing the inner product of two vectors.
3. Develop a sample program for collective operation MPI_Bcast.
4. Develop a sample program for collective operation MPI_Reduce.
5. Develop an implementation of collective operation MPI_Bcast using point-to-point communications. Compare the execution time of the developed version to the function MPI_Bcast.

TEST QUESTIONS

1. Select the correct statements:
   a. MPI is an acronym for My Personal identifier
   b. MPI is an acronym for Multiple Parallel Interface
   c. (+) MPI is an acronym for Message Passing Interface
   d. (+) Each parallel process in MPI has a number
   e. (+) MPI functions can be used only after MPI_Init was called
2. Select the correct statements:
   a. MPI_Comm_rank(&ProcRank) call defines the number of linearly independent rows of parallel processes incidence matrix
   b. MPI_Comm_rank(&ProcRank) call defines the rank (number) of the caller process
   c. (+) MPI_Comm_rank(&ProcRank, MPI_COMM_WORLD) call defines the rank (number) of the caller process
   d. MPI_Comm_size(&ProcNum) call defines the overall number of parallel processes started by an application
   e. (+) “MPI_Comm_size(&ProcNum, MPI_COMM_WORLD)” call defines the overall number of running parallel processes started by an application
3. A parallel program in MPI is:
   a. (+) A set of concurrently running processes

b. A set of concurrently running threads
c. A set of concurrently running processors

4. How the number of processes is determined during an MPI-program start?
   a. It will be equal to the number of nodes in cluster
   b. It is set through MPI capabilities in the source code directly
   c. (+) It is explicitly set during the MPI-program start
   d. It is set through a special environmental variable
5. It is required to measure the working time of a calculation block in an MPI program consisting of data input, data broadcasting, calculation block itself and results gathering. How one could do that correctly?
   a. Measure the time on the process with rank 0
   b. Measure the time on each process, and calculate the mean time.
   c. Measure the time on each process, and take maximum time.
   d. (+) Measure the time of calculation block start on every process and take the minimum of those, then measure the end of calculation block on every process and take the maximum of those, calculate difference between the two.
6. There is a need to distribute loop iterations between processes in an MPI program, assuming that execution times of iterations are nearly same.
   a. This will be done automatically among total number of processes
   b. Call MPI-function which determines loop iterations to be executed by each process
   c. (+) Divide the number of loop iterations by the number of processes. According to the rank of each process determine the starting and ending values of loop counter.
7. The minimal set of operations needed to organize informational communication between processors in systems with distributed memory includes only:
   a. (+) Data send/receive operations
   b. Data transfer operations and collective operations
   c. Collective operations only
8. Processes of a parallel MPI program:
   a. (+) May execute on separate processors, one processor can host several processes
   b. May execute on separate processors only
   c. Must execute on a single processor
9. Process’ number in MPI is called:
   a. (+) Process' rank
   b. Process' ID
   c. Process descriptor

10. Indication of the used communicator is:
    a. (+) Obligatory for all data transfer operations in MPI
    b. Optional for some date transfer operation in MPI
    c. Obligatory for some data transfer operations in MPI