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

04 Lecture
MPI Extensions

Brief description

Nizhni Novgorod
2014

04_LECTURE. MPI EXTENSIONS

OBJECTIVES

An objective of the lecture is to study nonblocking collective operations as well as mechanism of additional process creation and management after MPI program start.

ABSTRACT

Nonblocking collective operations, that was introduced in MPI-3 standard, are considered. The differences between blocking and nonblocking functions are discussed. The creation and management of processes during MPI program execution are overviewed. The connection mechanism to communicate with additionally created process is discussed.

BRIEF OVERVIEW

Nonblocking point-to-point data transmission functions, which were described in lecture 2, let combine the computations and the execution of the communication operations. Nonblocking collective operations, which were introduced in MPI-3 standard, combine the potential benefits of nonblocking point-to-point operations with the optimized implementation and message scheduling provided by collective operations. These operations are described in the first section of the lecture.

One way of doing this would be to perform a blocking collective operation in a separate thread, but onblocking collective communication often leads to better performance (avoids context switching, scheduler overheads, and thread management). Similarly to the blocking case, nonblocking collective operations are considered to be complete when the local part of the operation is finished (on root side or on other process side). Completion does not indicate that other processes have completed or even started the operation (unless otherwise implied by the description of the operation). Completion of a particular nonblocking collective operation also does not indicate completion of any other posted nonblocking collective (or send-receive) operations, whether they are posted before or after the completed operation.

Users should be aware that MPI implementations are allowed, but not required (with exception of MPI_IBARRIER), to synchronize processes during the completion of a nonblocking collective operation. Unlike point-to-point operations, nonblocking collective operations do not match with blocking collective operations. Once a process calls a collective operation, all other

processes in the communicator must eventually call the same collective operation, and no other collective operation with the same communicator in between.

At the second section of the lecture the dynamic process model, which allows for the creation and cooperative termination of processes after an MPI application has started, is considered.

Important classes of MPI applications which require process control: task farms, serial applications with parallel modules, problems that require a run-time assessment of the number and type of processes that should be started. MPI functions for creation additional process and establishing connections with existing processes provide a clean interface between an application and system software. MPI guarantees communication determinism in the presence of dynamic processes and maintains a consistent concept of a communicator, regardless of how its members came into existence. A communicator is never changed once created, and it is always created using deterministic collective operations.

The dynamic process model allows for the creation and cooperative termination of processes after an MPI application has started. It provides a mechanism to establish communication between the newly created processes and the existing MPI application. It also provides a mechanism to establish communication between two existing MPI applications, even when one did not "start" the other.

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. (1999). 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. (1999). Using MPI-2: Advanced Features of the Message Passing Interface (Scientific and Engineering Computation). – MIT Press.

EXERCISES

1. Develop a sample program for each method of nonblocking collective operations.
2. Develop a sample program using additional process in MPI program. Possible scheme to implement is “master-workers”.

TEST QUESTIONS

1. What types of collective operations exist in MPI?
   a. (+) Blocking
   b. (+) Nonblocking
   c. Synchronous
   d. Asynchronous
2. Usage of nonblocking collective operations means that
   a. Data transmission will be finished immediately after calling nonblocking operation.
   b. (+) Nonblocking function will be finished immediately after calling
3. Usage of nonblocking collective operations means that
   a. Buffer that is used in operation may be changed after calling the nonblocking function.
   b. (+) Buffer that is used in operation may be changed after data transmission completion.
4. Completion the nonblocking collective operation on some process indicates that
   a. Other processes have completed the operation to
   b. Other processes have started the operation
   c. (+) Condition of operation on other processes is unknown.
5. May nonblocking collective operations match with blocking ones?
   a. Yes, like point-to-point operations.

b. (+) No, unlike point-to-point operations.

6. To make a nonblocking broadcast one should use
   a. MPI_Bcast
   b. (+) MPI_Ibcast
   c. MPI_Scatter
   d. MPI_Iscatter
7. To make a nonblocking reduction one should use
   a. MPI_Bcast
   b. MPI_Ibcast
   c. MPI_Reduce
   d. (+) MPI_Ireduce
8. To make a nonblocking distribution one should use
   a. MPI_Bcast
   b. MPI_Ibcast
   c. MPI_Scatter
   d. (+) MPI_Iacatter
9. To make a nonblocking gathering data on each process of communicator one should use
   a. MPI_Ialltoall
   b. MPI_IallReduce
   c. (+) MPI_Iallgather
10. To make a nonblocking reduction on each process of communicator one should use
    a. MPI_Ialltoall
    b. (+) MPI_IallReduce
    c. MPI_Iallgather
11. The dynamic process model provides a mechanism to establish communication
    a. (+) Between the newly created processes and the existing MPI application
    b. Between “master” and “workers”.
    c. Between server and clients.
    d. (+) Between two existing MPI applications, even when one did not “start” the other.