



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

Test questions

Nizhni Novgorod
2014

01_LECTURE. THE FUNDAMENTALS OF MPI

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 data transfer operation in MPI
 - c. Obligatory for some data transfer operations in MPI

02_LECTURE. COLLECTIVE AND POINT-TO-POINT COMMUNICATIONS

11. Select correct statements:
 - a. All application processes participate in collective operations
 - b. (+) Only processes of some communicator participate in collective operations
 - c. (+) A function corresponding to a collective operation must be called by every process, possibly with its own parameters
 - d. (+) MPI_Barrier is a collective operation
 - e. MPI_Send is a collective operation
12. Collective data transfer operations must include as participants:
 - a. (+) All processes of the used communicator
 - b. Some of the processes of the used communicator
 - c. All processes of used group of processes
13. MPI collective operations:
 - a. (+) May be implemented using point-to-point operations, but most likely such solution will be ineffective
 - b. Cannot be implemented using point-to-point operations in principle
 - c. Can be implemented using point-to-point operation, but not fully
14. Collective operations in MPI are:
 - a. (+) Data transfer operations having all processes of used communicator as participants
 - b. Operations on groups of processes
 - c. Operations on communicators
15. MPI_Reduce data reduction operation can be described as:
 - a. (+) A data transfer operation in which collected values are processed during transfer and the result of processing is given to the root process only
 - b. A data transfer operation in which collected values are processed and partial values of reduction results are given to all processes of the parallel program
 - c. A data transfer operation, in which collected results are processed in some way, and the result of processing is given to all processes
16. Data broadcasting operation is:
 - a. (+) An operation when the root process sends values to other processes and all processes receive all the data that was sent
 - b. An operation when root process sends values to other processes and all processes receive only part of the original data
 - c. An operation when the root process sends different values to other processes
17. Data transfer operations in MPI include
 - a. (+) point-to-point and collective operations
 - b. paired and group operations
 - c. individual and collective operations
18. In the Synchronous mode of point-to-point data transmission the function of data sending is terminated
 - a. (+) when the confirmation of the reception beginning for the transmitted message comes from the receiving process
 - b. if the message reception operation has already been initiated
 - c. after the message has been copied in the system buffer

19. In the Buffered mode of point-to-point data transmission the function of data sending is terminated:
- a. (+) after the message has been copied in the system buffer
 - b. when the confirmation of the reception beginning for the transmitted message comes from the receiving process
 - c. if the message reception operation has already been initiated
20. The Ready mode of point-to-point data transmission the function of data sending may be used only:
- a. (+) if the message reception operation has already been initiated
 - b. if the size of the message is less than the size of the system buffer
 - c. if the message reception operation will be initiated after data sending start

03_LECTURE. DERIVED DATA TYPES, COMMUNICATORS AND VIRTUAL TOPOLOGIES

21. The derived data type in MPI is
 - a. (+) The description of a set of the values of the predetermined in MPI type. The described values are not necessarily located continuously in the memory.
 - b. The sequence of data value descriptions in the type. Each value is described using displacement from some base address.
 - c. The description of data set of base algorithmic language types in terms of this language itself.
22. The type signature of the derived types in MPI is
 - a. (+) The part of the type map, which contains only the types of values.
 - b. The part of the type map, which contains only the displacements of values.
 - c. The size of memory in bytes necessary for the storing of one element from the type.
23. The extent of the derived types in MPI is
 - a. (+) The amount of memory in bytes that should be allocated for a derived type element.
 - b. The number of bytes occupied by the data in a derived type element only.
 - c. Displacement of the first byte of a derived type.
24. The derived type in the vector constructing method is
 - a. (+) A number of blocks of the initial type elements with regular interval in the memory between blocks.
 - b. Continuous sequence of elements of the initial type.
 - c. A set of blocks of different sizes of the initial type elements and memory space between blocks may be different.
25. The derived type in the index constructing method is
 - a. (+) A set of blocks of different sizes of the initial type elements and memory space between blocks may be different.
 - b. A number of blocks of the initial type elements with regular interval in the memory between blocks.
 - c. Continuous sequence of elements of the initial type.
26. What is the difference between vector and H-vector (index and H-index) derived type?
 - a. (+) Intervals between blocks are set in bytes (not in the initial data type elements).
 - b. H-vector (h-index) derived type may contain a sequence of elements of the initial data type with regular interval between them.
 - c. H-vector (h-index) derived type may contain a sequence of elements of the initial data type with different interval between them.
27. The derived type in the structural constructing method is
 - a. (+) The most general constructing way for creating the derived data type, when the corresponding type map is set directly.
 - b. A number of blocks of the initial type elements with regular interval in the memory between blocks.
 - c. Continuous sequence of elements of the initial type.
 - d. A set of blocks of different sizes of the initial type elements and memory space between blocks may be different.
28. What virtual topologies are supported in MPI?
 - a. (+) A rectangular grid with any dimension.

- b. A complete graph only.
 - c. (+) A graph with any structure.
29. Cartesian topology is the presentation of a set of processes as
- a. (+) Rectangular grid.
 - b. A complete graph.
 - c. A graph with any structure.
30. What topology the initial communicator MPI_COMM_WORLD has?
- a. Rectangular grid.
 - b. (+) A complete graph.
 - c. A graph with any structure.

04_LECTURE. MPI EXTENSIONS

31. What types of collective operations exist in MPI?
 - a. (+) Blocking
 - b. (+) Nonblocking
 - c. Synchronous
 - d. Asynchronous
32. 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
33. 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.
34. 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.
35. May nonblocking collective operations match with blocking ones?
 - a. Yes, like point-to-point operations.
 - b. (+) No, unlike point-to-point operations.
36. To make a nonblocking broadcast one should use
 - a. MPI_Bcast
 - b. (+) MPI_Ibcast
 - c. MPI_Scatter
 - d. MPI_Iscatter
37. To make a nonblocking reduction one should use
 - a. MPI_Bcast
 - b. MPI_Ibcast
 - c. MPI_Reduce
 - d. (+) MPI_Ireduce
38. To make a nonblocking distribution one should use
 - a. MPI_Bcast
 - b. MPI_Ibcast
 - c. MPI_Scatter
 - d. (+) MPI_Iacatter
39. To make a nonblocking gathering data on each process of communicator one should use
 - a. MPI_Ialltoall
 - b. MPI_IallReduce
 - c. (+) MPI_Iallgather
40. To make a nonblocking reduction on each process of communicator one should use
 - a. MPI_Ialltoall
 - b. (+) MPI_IallReduce
 - c. MPI_Iallgather
41. 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.

05_PRACTICE. PARALLEL ALGORITHMS OF MATRIX-VECTOR MULTIPLICATION

42. What is the complexity order for matrix-vector multiplication?
- $O(n)$
 - (+) $O(n^2)$
 - $O(n^3)$
43. What are the main methods of distributing matrix elements among processors?
- (+) Partitioning data into stripes (vertically and horizontally).
 - Partitioning data using LU decomposition.
 - (+) Partitioning data into rectangular fragments (blocks).
44. Why is it admissible to duplicate the vector-operand to all the processors in developing a parallel algorithm of matrix-vector multiplication?
- (+) According to computational scheme each processor should have the vector.
 - Because time cost of this operation is very small.
 - Because data broadcast works faster than data scattering.
45. What number of processes may be used during the execution of parallel implementation of matrix-vector multiplication?
- (+) Any possible number of processes may be used.
 - The number of processes should be a perfect square.
 - The number of processes should be equal the number of matrix rows.
46. For the efficient implementation of parallel matrix-vector multiplication algorithm based on block matrix decomposition, it is necessary that the block sizes would:
- (+) be approximately equal to the size of computing core cache,
 - be smaller in height than in width,
 - fit in width to the cache-line
47. For the efficient implementation of parallel matrix-vector multiplication algorithm based on block-striped horizontal matrix decomposition, it is necessary that the stripe height would:
- be as greater as possible,
 - (+) no matter in the case of great data volume,
 - be as greater as possible in case of small matrices and be as smaller as possible in case of great ones
48. What function should be used to distribute between processes the matrix size in parallel matrix-vector multiplication?
- MPI_Send
 - (+) MPI_Bcast
 - MPI_Scatter
 - MPI_Gather
49. What function should be used to distribute between processes the matrix in parallel matrix-vector multiplication?
- MPI_Send
 - MPI_Bcast
 - (+) MPI_Scatter
 - MPI_Gather
50. What function should be used to get the result vector in parallel matrix-vector multiplication?
- MPI_Reduce

- b. MPI_Bcast
 - c. MPI_Scatter
 - d. (+) MPI_Gather
51. What virtual topology should be used to implement parallel matrix-vector multiplication?
- a. (+) MPI_COMM_WORLD is enough.
 - b. Cartesian topology
 - c. Special graph topology

06_PRACTICE. PARALLEL ALGORITHMS OF MATRIX MULTIPLICATION

52. What is the complexity order for matrix-matrix multiplication?
- $O(n)$
 - $O(n^2)$
 - (+) $O(n^3)$
53. What are the main methods of distributing matrix elements among processors?
- (+) Partitioning data into stripes (vertically and horizontally).
 - Partitioning data using LU decomposition.
 - (+) Partitioning data into rectangular fragments (blocks).
54. What number of processes may be used during the execution of parallel implementation of matrix multiplication (partitioning data into stripes)?
- (+) Any possible number of processes may be used.
 - The number of processes should be a perfect square.
 - The number of processes should be equal the number of matrix rows.
55. What number of processes may be used during the execution of parallel implementation of matrix multiplication (Fox method)?
- Any possible number of processes may be used.
 - (+) The number of processes should be a perfect square.
 - The number of processes should be equal the number of matrix rows.
56. What function should be used to distribute between processes the matrix size in parallel matrix multiplication?
- MPI_Send
 - (+) MPI_Bcast
 - MPI_Scatter
 - MPI_Gather
57. What function should be used to distribute between processes the matrix in parallel matrix multiplication?
- MPI_Send
 - MPI_Bcast
 - (+) MPI_Scatter
 - MPI_Gather
58. What function should be used to perform the cyclic shift of blocks of the matrix B to implement parallel matrix multiplication?
- MPI_Send
 - MPI_Bcast
 - MPI_Scatter
 - MPI_Gather
 - (+) MPI_Sendrecv_replace
59. What function should be used to get the result matrix in parallel matrix multiplication?
- MPI_Reduce
 - MPI_Bcast
 - MPI_Scatter
 - (+) MPI_Gather
60. What virtual topology should be used to implement parallel matrix multiplication?
- MPI_COMM_WORLD is enough.

- b. (+) Cartesian topology
 - c. Special graph topology
61. What algorithm may be used to implement parallel matrix multiplication?
- a. (+) Fox method
 - b. (+) Cannon method
 - c. Gauss method

07_PRACTICE. PARALLEL METHODS OF SOLVING THE LINEAR EQUATION SYSTEMS

62. What is the complexity order of the Gaussian elimination method when applying to triangular matrix?
- $O(n)$
 - (+) $O(n^2)$
 - $O(n^3)$
63. What is the complexity order of the Gaussian elimination method when applying to square matrix?
- $O(n)$
 - $O(n^2)$
 - (+) $O(n^3)$
64. What is the relation between the error of the Gaussian method without choosing the pivot element and the error of the Gaussian method with choosing the pivot element by column?
- The errors are the same and they are comparable to rounding error
 - The error of the common Gaussian method is smaller than the error of the method with choosing the pivot element
 - (+) The error of the Gaussian method with choosing the pivot element is smaller than of the common method.
65. What is the relation between the error of the Gaussian method without choosing the pivot element and the error of the Gaussian method with choosing the pivot element by column and by row?
- The errors are the same and they are comparable to rounding error
 - The error of the common Gaussian method is smaller than of the method with choosing the pivot element
 - (+) The error of the Gaussian method with choosing the pivot element is smaller than of the common method.
66. What number of processes may be used during the execution of parallel implementation of Gaussian method?
- (+) Any possible number of processes may be used.
 - The number of processes should be a perfect square.
 - The number of processes should be equal the number of matrix rows.
67. What function should be used to distribute between processes the matrix size in parallel Gaussian method?
- MPI_Send
 - (+) MPI_Bcast
 - MPI_Scatter
 - MPI_Gather
68. What function should be used to distribute between processes the matrix in parallel Gaussian method?
- MPI_Send
 - MPI_Bcast
 - (+) MPI_Scatter
 - MPI_Gather

69. What function should be used to get the result vector in parallel Gaussian method (Gaussian elimination stage)?
- (+) `MPI_AllReduce`
 - `MPI_Bcast`
 - `MPI_Scatter`
 - `MPI_Gather`
70. What function should be used to get the result vector in parallel Gaussian method (back substitution stage)?
- `MPI_AllReduce`
 - `MPI_Bcast`
 - `MPI_Scatter`
 - (+) `MPI_Gather`
71. What virtual topology should be used to implement parallel Gaussian method?
- (+) `MPI_COMM_WORLD` is enough.
 - Cartesian topology
 - Special graph topology

08_PRACTICE. PARALLEL ALGORITHMS OF GRAPH PROCESSING

72. What data structure may be used to present dense weighted graph?
 - a. (+) Adjacency matrix
 - b. Stack
 - c. Queue
73. What is the complexity order for Floyd algorithm for solving the problem of search for the shortest paths?
 - a. $O(n)$
 - b. $O(n^2)$
 - c. (+) $O(n^3)$
74. What are the main methods of distributing the adjacency matrix elements among processors?
 - a. (+) Partitioning data into stripes (vertically and horizontally).
 - b. Partitioning data using LU decomposition.
 - c. (+) Partitioning data into rectangular fragments (blocks).
75. What number of processes may be used during the execution of parallel implementation of Floyd algorithm for solving the problem of search for the shortest paths?
 - a. (+) Any possible number of processes may be used.
 - b. The number of processes should be a perfect square.
 - c. The number of processes should be equal the number of matrix rows.
76. What function should be used to distribute between processes the adjacency matrix size in parallel implementation of Floyd algorithm for solving the problem of search for the shortest paths?
 - a. MPI_Send
 - b. (+) MPI_Bcast
 - c. MPI_Scatter
 - d. MPI_Gather
77. What function should be used to distribute between processes the adjacency matrix in parallel implementation of Floyd algorithm for solving the problem of search for the shortest paths?
 - a. MPI_Send
 - b. MPI_Bcast
 - c. (+) MPI_Scatter
 - d. MPI_Gather
78. What function should be used to get the result vector in parallel implementation of Floyd algorithm for solving the problem of search for the shortest paths?
 - a. MPI_Reduce
 - b. MPI_Bcast
 - c. MPI_Scatter
 - d. (+) MPI_Gather
79. What virtual topology should be used to implement parallel implementation of Floyd algorithm for solving the problem of search for the shortest paths?
 - a. (+) MPI_COMM_WORLD is enough.
 - b. Cartesian topology
 - c. Special graph topology
80. What method may be used to check the correctness of the parallel implementation of Floyd algorithm for solving the problem of search for the shortest paths?

- a. (+) Compare the results of the parallel program with the results of the serial one.
- b. (+) Implement algorithm that calculate the path weight for every possible path in the graph and compare the results.
- c. The implementation of the parallel program guarantees the correctness.