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

05 Practice
Parallel Algorithms of Matrix-Vector Multiplication

Brief description

Nizhni Novgorod
2014

05_PRACTICE. PARALLEL ALGORITHMS OF MATRIX-VECTOR MULTIPLICATION

OBJECTIVES

An objective of the practice is to demonstrate a practical application of the parallel linear algebra algorithms by example of the matrix-vector multiplication problem.

ABSTRACT

The work is organized in the following way. The matrix-vector multiplication problem is stated. Implementation of serial solving method is discussed and demonstrated. Possible parallel algorithm and scheme of data distribution are considered. Implementation of parallel algorithm using MPI is described.

BRIEF OVERVIEW

Matrices and matrix operations are widely used in mathematical modeling of various processes, phenomena and systems. Matrix calculations are the basis of many scientific and engineering calculations. Computational mathematics, physics, economics are only some of the areas of their application.

As the efficiency of carrying out matrix computations is highly important many standard software libraries contain procedures for various matrix operations. The amount of software for matrix processing is constantly increasing. New efficient storage structures for special type matrix (triangle, banded, sparse etc.) are being created. Highly efficient machine-dependent algorithm implementations are being developed. The theoretical research into searching faster matrix calculation method is being carried out.

Being highly time consuming, matrix computations are the classical area of applying parallel computations. On the one hand, the use of highly efficient multiprocessor systems makes possible to substantially increase the complexity of the problem solved. On the other hand, matrix operations, due to their rather simple formulation, give a nice opportunity to demonstrate various techniques and methods of parallel programming.

In this practice the parallel programming methods for matrix-vector multiplication are discussed. It is assumed, that the matrices are dense, i.e. the number of zero elements in them is insignificant in comparison to the general number of matrix elements.

The repetition of the same computational operations for different matrix elements is typical of different matrix calculation methods.

Because data parallelism exists in this case, the problem to parallelize matrix operations can be reduced in most cases to matrix distributing among the processors of the computer system. The choice of matrix distribution method determines the use of the definite parallel computation method. The availability of various data distribution schemes generates a range of parallel algorithms of matrix computations.

The most general and the most widely used matrix distribution methods consist in partitioning data into stripes (vertically and horizontally) or rectangular fragments (blocks). Partitioning into stripes is used in this practice.

The first section of the practice contains the matrix-vector multiplication problem statement and pseudocode of the algorithm.

In the second section the project for Microsoft Visual Studio is developed step-by-step. The developed application implements the serial algorithm as well as the necessary steps to input initial data (matrix and vector), finish the execution correctly, and carry out the computational experiments.

In the third section the data distribution scheme is considered, the computation organization is presented, so parallel algorithm is formulated.

The last section is devoted to implementation of previously described parallel algorithm as an MPI parallel program. Serial implementation is used as the basis. Parallel program is developed step-by-step like serial one. Necessary steps include parallel program initialization, data input (matrix and vector), data distribution, parallel matrix-vector multiplication, gathering the results.

FOR STUDENTS

Matrix-vector multiplication problems are often used as a demonstrative example in parallel programming. Therefore, they are widely used in the literature. We recommend the papers by Quinn (2004) and Kumar, et al. (2003) as an additional educational material. The wide discussion of the parallel realization of matrix computations is given in Dongarra (1999).

REFERENCES

1. Dongarra, J.J., Duff, L.S., Sorensen, D.C., Vorst, H.A.V. (1999). Numerical Linear Algebra for High Performance Computers (Software, Environments, Tools). Soc for Industrial & Applied Math.

2. Quinn, M.J. (2004). Parallel Programming in C with MPI and OpenMP. – New York, NY: McGraw-Hill.
3. Foster, I. (1995). Designing and Building Parallel Programs: Concepts and Tools for Software Engineering. Reading, MA: Addison-Wesley.

EXERCISES

1. Study the parallel algorithm of matrix-vector multiplication based on column-wise block-striped matrix partitioning. Develop a program implementation of this algorithm.
2. Study the parallel algorithm of matrix-vector multiplication based on chessboard block matrix partitioning. Develop a program implementation of this algorithm.

TEST QUESTIONS

1. What is the complexity order for matrix-vector multiplication?
   a. O(n)
   b. (+) $O(n^2)$
   c. $O(n^3)$
2. What are the main methods of distributing 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).
3. Why is it admissible to duplicate the vector-operand to all the processors in developing a parallel algorithm of matrix-vector multiplication?
   a. (+) According to computational scheme each processor should have the vector.
   b. Because time cost of this operation is very small.
   c. Because data broadcast works faster than data scattering.
4. What number of processes may be used during the execution of parallel implementation of matrix-vector multiplication?
   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.
5. For the efficient implementation of parallel matrix-vector multiplication algorithm based on block matrix decomposition, it is necessary that the block sizes would:
   a. (+) be approximately equal to the size of computing core cache,
   b. be smaller in height than in width,

   c. fit in width to the cache-line

6. 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:
   a. be as greater as possible,
   b. (+) no matter in the case of great data volume,
   c. be as greater as possible in case of small matrices and be as smaller as possible in case of great ones
7. What function should be used to distribute between processes the matrix size in parallel matrix-vector multiplication?
   a. MPI_Send
   b. (+) MPI_Bcast
   c. MPI_Scatter
   d. MPI_Gather
8. What function should be used to distribute between processes the matrix in parallel matrix-vector multiplication?
   a. MPI_Send
   b. MPI_Bcast
   c. (+) MPI_Scatter
   d. MPI_Gather
9. What function should be used to get the result vector in parallel matrix-vector multiplication?
   a. MPI_Reduce
   b. MPI_Bcast
   c. MPI_Scatter
   d. (+) MPI_Gather
10. 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