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"

INTRODUCTION TO PARALLEL PROGRAMMING

Lectures 7, 8. Parallel Methods for Matrix Multiplication
for Systems with Shared Memory

Nizhni Novgorod

2014

Lecture_7, 8_. Parallel Methods for Matrix Multiplication for systems with shared memory

Matrix multiplication is one of the essential problems in matrix calculations. This Section discusses several parallel algorithms for carrying out the operation. Two of them are based on block striped data decomposition. The other two use checkerboard block data decomposition, the latter being based on matrix decomposition into blocks sized so that they can be fully cached.

Multiplying an $m \times n$ matrix *A* with *m* rows and *n* columns and an $n \times l$ matrix *B* with *n* rows and *l* columns produces an $m \times l$ matrix *C* with *m* rows and *l* columns. Each element of the matrix *C* is calculated according to the formula:

$$c_{ij} = \sum_{k=0}^{n-1} a_{ik} \cdot b_{kj}, 0 \le i < m, 0 \le j < l. \qquad (7.1)$$

As it can be seen in (7.1), each element of the matrix *C* is the result of the inner product of the corresponding row of the matrix *A* and column of the matrix *B*:

$$c_{ij} = \left(a_i, b_j^T\right), a_i = \left(a_{i0}, a_{i1}, ..., a_{in-1}\right), b_j^T = \left(b_{0j}, b_{1j}, ..., b_{n-1j}\right)^T. \qquad (7.2)$$

This algorithm executes $m \cdot n \cdot l$ multiplications and the same number of additions of the initial matrix elements. In case of square matrices, the size of which is $n \times n$, the number of the executed operations is the order $O(n^3)$. There are also sequential matrix multiplication algorithms of smaller computational complexity (for instance, the Strassen algorithm). But studying these algorithms though requires certain efforts and for simplicity we will use the above described sequential algorithm as the basis for parallel method development in this section. We will also assume further that all matrices are square and their sizes are $n \times n$.

This lecture describes four parallel methods of matrix multiplication. The first and the second algorithms are based on block striped matrix decomposition. The first algorithm version is based on distribution of one of the argument matrices (matrix *A*) and the resulting matrix among the parallel program threads. For the purposes of the second algorithm, the first matrix decomposition into horizontal stripes and the second one is decomposed into vertical stripes with each parallel program thread computing one block of the resulting matrix *C*. This approach is implemented based on nested parallelism mechanisms. The lecture also describes two checkerboard matrix multiplication algorithms, the latter being based on matrix decomposition into blocks sized so that they can be fully cached.

See the general diagram in Fig. 7.1 for speedup values obtained in the course of computational experiments for all the described algorithms (parallel algorithm speedup is shown in relation to the original sequential method of matrix multiplication).

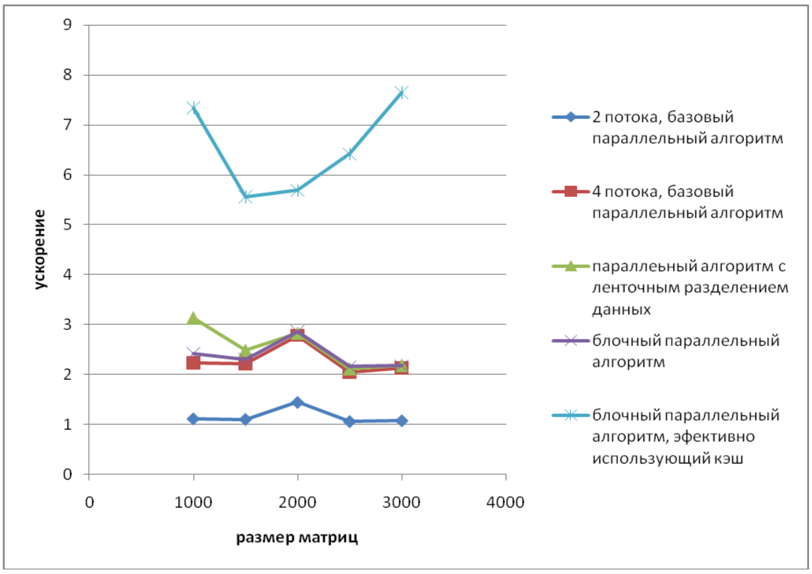

Fig. 7.1. Matrix multiplication speedup for the four parallel algorithms described in this lecture

Test questions

1. What is the statement of the matrix multiplication problem?
2. Give the examples of the problems, which make use of the matrix multiplication operations.
3. Give the examples of various sequential algorithms of matrix multiplication operations. Is the complexity various in case of different algorithms?
4. What methods of data decomposition are used in developing parallel algorithms of matrix multiplication?
5. Give general schemes of the described matrix multiplication algorithms.
6. What information communications are carried out for the algorithms in case of the block-striped data decomposition?
7. What information communications are carried out for the algorithms in case of the checkerboard data decomposition?
8. Which of the described algorithms has the best parameters in terms of speedup and efficiency?

9. Estimate the possibility of matrix multiplication as a sequence of vector-matrix multiplication operations.
10. What OpenMP functions proved to be necessary for implementation of parallel algorithms?

Practice

1. Implement checkerboard block matrix multiplication algorithms that could be executed for rectangular grids of general streams.

2. Implement matrix multiplication using the already developed matrix-vector multiplication programs.

References

1. **Kumar V.**, Grama A., Gupta A., Karypis G. Introduction to Parallel Computing , Inc. 1994 (2th edition, 2003).
2. **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/
3. **Blackford,** L. S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J. J., Hammarling, S., Henry, G., Petitet, A., Stanley, D. Walker, R.C. Whaley, K. (1997). Scalapack Users' Guide (Software, Environments, Tools). Soc for Industrial & Applied Math.
4. **Foster**, I. (1995). Designing and Building Parallel Programs: Concepts and Tools for Software Engineering. Reading, MA: Addison-Wesley.
5. **Andrews**, G. R. (2000). Foundations of Multithreaded, Parallel, and Distributed Programming.. – Reading, MA: Addison-Wesley
6. **Kahaner, D**., Moler, C., Nash, S. (1988). Numerical Methods and Software. – Prentice Hall
7. **Quinn**, M. J. (2004). Parallel Programming in C with MPI and OpenMP. – New York, NY: McGraw-Hill.
8. **Wilkinson**, B., Allen, M. (1999). Parallel programming. – Prenrice Hall.