

Optimization of the amount of transmitted data in parallel algorithms for iterative methods with a triangular preconditioner

Asya Atayan*, Elena Rakhimbaeva, and Alexander Chistyakov

Don State Technical University, Rostov-on-Don, 344000, Russia

Abstract. Software has been developed in the C++ programming language using the MPI parallel programming technology, designed for mathematical modeling of the transport of substances in coastal systems. When calculating the dynamics of the spread of a pollutant, the decomposition of the computational domain was carried out to organize the computational process on a multiprocessor computer system K-60 in KIAM RAS. To solve the system of grid equations obtained as a result of the approximation of the problem, iterative methods were used, with a triangular preconditioner.

1 Introduction

Today, in the modern world, the issues of ecology and the preservation of the quality of coastal (especially fresh) and commercial waters are becoming more and more relevant. Environmental problems are associated not only with climate change, loss of biological diversity, but also with an increase in environmental damage from natural disasters and man-made disasters that affect water bodies and their inhabitants. To preserve water complexes, maintain their integrity and life-supporting functions, it is important not only to take organizational, engineering and technical solutions, but also to have highly effective methods for modeling various potential and actual mechanisms of primary and secondary pollution of coastal systems, which make it possible to quickly and efficiently based on interrelated high-precision models hydrophysics and hydrobiology to predict the processes of the spread of pollution and the occurrence of hazardous phenomena in coastal systems [1, 2].

To increase the performance of mathematical models based on solving diffusion-convection problems, it is necessary to include factors that have a significant impact on hydrobiological processes: parameterizable microturbulent diffusion and advective transport in various directions [3]. The calculation of data on a multiprocessor computer system can significantly reduce the computation time.

* Corresponding author: atayan24@mail.ru

2 Decomposition of the computational domain in one spatial direction

Let us describe a method for constructing a parallel algorithm for solving the problem of pollutant transport in the two-dimensional case. The computational domain is covered with a uniform rectangular grid [4]:

$$\omega_{i,j} : \{x(ih_x) = x_i, y(jh_y) = y_j; i \in \overline{0, N_x - 1}, j \in \overline{0, N_y - 1}; h_x N_x = l_x, h_y N_y = l_y\}, \quad (1)$$

where i, j are the indices of the computational domain, h_x, h_y are the steps in spatial directions, N_x, N_y are the number of steps in the spatial directions, l_x, l_y are the characteristic dimensions of the computational domain.

At the nodes of the computational grid, the values of the water flow velocity field are calculated $u(x, y) : u_{i,j}$ at $i \in \overline{1, N_x - 2}, j \in \overline{1, N_y - 2}$, while along the perimeter ($i \in \overline{0, N_x - 1}, j \in \overline{0, N_y - 1}$) there are fictitious nodes. Let us decompose the computational domain along the spatial direction by Oy straight lines parallel to the axis Ox , and denote the w^r subdomain with the number r , $0 \leq r \leq R-1$, where R is the number of subdomains into which the original domain is divided. The calculated nodes of the region w^r are the elements $u_{i,j}^r$ at $i \in \overline{1, N_x - 2}, j \in \overline{1, N_2^r - 2}$. The partition of the original region is made in such a way that adjacent regions w^r and w^{r+1} intersect at two nodes along the direction perpendicular to the partition lines and equalities take place $u_{i, N_2^r - 2}^r = u_{i, 0}^{r+1}, u_{i, N_2^r - 1}^r = u_{i, 1}^{r+1}$ (Fig. 1, arrows show fictitious nodes.).

To represent the field value $u(x, y)$ in vector form, a pair of indices i, j can be associated with a value m that describes the ordinal number of the elements of the vector $u : m = i + jN_x, 0 \leq m \leq n-1$, where n is the length of the vector $u = (u_0, u_1, \dots, u_{n-1})^T$. This representation is convenient to use when describing and studying algorithms for solving grid equations by iterative methods.

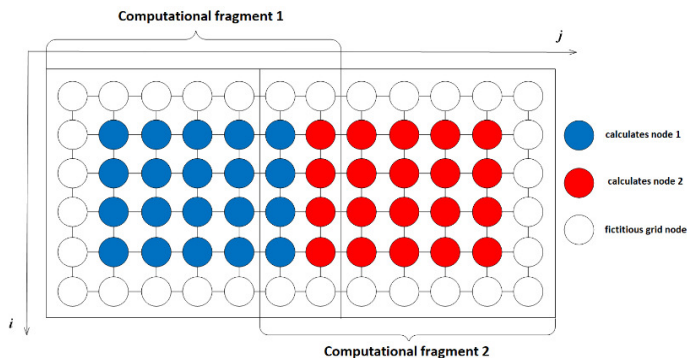


Fig. 1. Decomposition of the computational domain

For fragments w^r , obtained as a result of decomposition of the computational domain in one spatial direction, it is necessary to know two parameters: the initial index $j = N_1^r$ in the initial computational domain and the width of the fragment N_2^r . The index number N_1^r ,

from which the corresponding fragment of the computational domain begins, can be calculated using the formula:

$$N_1^r = \lfloor r \cdot (N_y - 2) / R \rfloor, \quad (2)$$

where $\lfloor x \rfloor$ – the «floor» function is defined as the largest integer less than or equal to x , $\lceil x \rceil$ – the «ceiling» function is defined as the smallest integer greater than or equal to x .

The width of the subregion w^r along the axis Oy is calculated by the formula:

$$N_2^r = \lceil (r + 1) \cdot (N_y - 2) / R \rceil - N_1^r + 2. \quad (3)$$

The calculation of data on a multiprocessor computer system can significantly reduce the computation time. However, the time efficiency of a computing system may not always be expected. In this case, it is correct to carry out a theoretical analysis of the calculation of the computation time based on regression analysis [5-8].

Figure 2 shows a graph of the dependence of the transfer time on the amount of data for a different number of exchanges between the nodes of the computer system. The graph shows that the transfer time dependency function has a jump when the amount of data transferred is approximately 512 floating point numbers. Let's denote this value $N_{\max} = 512$.

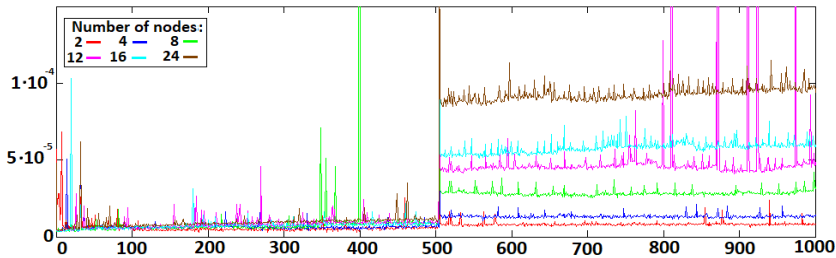


Fig. 2. Dependence of data transfer time on volume when working with a different number of computing nodes

The following parameters are usually used to theoretically evaluate the operation of computing systems:

- t_a , the execution time of one arithmetic operation;
- t_l , time of organization of data transmission (latency);
- t_x , the transmission time of one dat.

3 Calculation of latency time based on the least squares method

Let there be some variable i , which represents the i -th observation of the dependent variable y_i , and let's denote the explanatory factors by the vector x_i . Then the multiple regression model can be represented as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \varepsilon_i, \quad (4)$$

where β is a free term; ε_i – the member containing the error; $i = 1, 2, \dots, p$.

A finite vector of dimension n is a matrix of values of explanatory factors, dimension n on the $(p+1)$. The model in matrix form will look like

$$Y = X\beta + \varepsilon. \quad (5)$$

The estimate of this model for some sample will be the equation in which $\beta = (\beta_0 \beta_1 \dots \beta_p)^T$, $\varepsilon = (\varepsilon_1 \varepsilon_2 \dots \varepsilon_n)^T$. To estimate the vector of unknown parameters β , you can use the least squares method.

The condition for minimizing the residual sum of squares can be represented as:

$$S = \sum_{i=1}^n (y_{x_i} - y_i)^2 = \sum_{i=1}^n \varepsilon_i^2 = \varepsilon\varepsilon^T = (Y - X\beta)^T (Y - X\beta) \rightarrow \min. \quad (6)$$

Performing transformations in (6) we obtain

$$S = Y^T Y - \beta^T X^T Y - Y^T X \beta + \beta^T X^T X \beta. \quad (7)$$

The product $Y^T X \beta$ is, as a result, a certain matrix with dimension $1 \cdot n [n(p+1)] \cdot [l(p+1)]$. This implies

$$S = Y^T Y - 2\beta^T X^T Y - \beta^T X X \beta \rightarrow \min, \quad (8)$$

where $X^T X$ is the matrix of sums of first powers, squares, and pairwise products of n observations of explanatory factors; $X^T Y$ is the vector of products, dimension n , of observations of explanatory factors and dependent variables.

The solution of the matrix equation will be the vector $\beta = (X^T X)^{-1} X^T Y$, where $(X^T X)^{-1}$ is the matrix inverse to the matrix of system coefficients; $X^T Y$ is the vector of its - free members.

Knowing the vector β , any multiple regression equation can be represented as $y_{x_i} = X_i^T \beta$. To calculate the operating time of the computing system y_i , it acts as the final time, and the explanatory factors indicated by the vector x_i are: the size of the computational grid, the number of computing nodes used. Thus, it seems possible to calculate the average running time of the entire system. Based on the presented regression analysis, a linear dependence of the operating time of a software module that implements a parallel algorithm was obtained on the amount of transmitted data and the number of involved computing nodes of a multiprocessor system (Fig. 3) for cases where the amount of transmitted data is less than (Fig. 3a) and more than 512 elements (Fig. 3b).

Latency and data transfer times are calculated using the least squares method. The formula for latency is:

$$t_l(R) = \begin{cases} 5,21 \times 10^{-6} + 1,53 \times 10^{-7} (R-2), & \text{если } N_x \leq N_{\max}, \\ 6,733 \times 10^{-6} (R-2), & \text{если } N_x > N_{\max}. \end{cases} \quad (9)$$

Transmission time per data $t_x = 3,3 \times 10^{-9}$ [c].

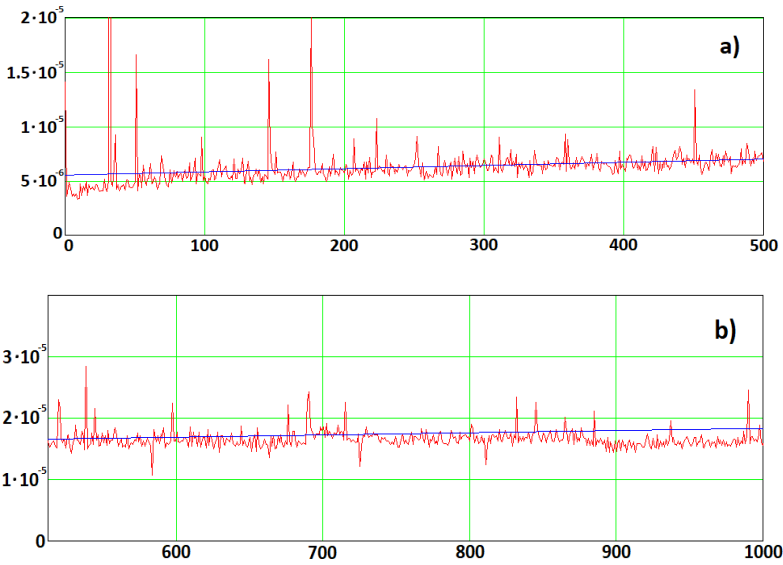


Fig. 3. Dependence of data transfer time on volume a) data volume up to 512 elements; b) data volume more than 512 elements

4 Iterative methods for solving grid equations

Consider the system of linear algebraic equations:

$$Ax = f, \tag{10}$$

where $A = (a_{ij})_{i,j=1}^n$ is a matrix that has an inverse matrix, $x = (x_1, x_2, \dots, x_n)^T$ and $f = (f_1, f_2, \dots, f_n)^T$ are the column vectors of the unknown and right parts, respectively.

To study the convergence of iterative methods, that is, to establish the validity of the equality $\lim_{n \rightarrow \infty} \|x^n - x\| = 0$, where x^n is the exact solution, it is advisable to write these methods in matrix rather than coordinate form.

Represent the matrix A as the sum of three matrices: $A = A^- + D + A^+$. Obviously, the Jacobi method, using the introduced notation, in vector form takes the form: $x^{n+1} = -D^{-1}(A^+ + A^-)x^n + D^{-1}f$, where D^{-1} is the matrix inverse to the matrix D : $D^{-1} \cdot D = E$, E – identity matrix, D – diagonal matrix.

Represent the Jacobi method as follows:

$$D(x^{n+1} - x^n) + Ax^n = f. \tag{11}$$

The representation of the Seidel method in vector form is:

$$(D + A^-)(x^{n+1} - x^n) + Ax^n = f. \tag{12}$$

These vector equalities are special cases of the canonical form of one-step (two-layer) iterative schemes of the form:

$$B \frac{x^{n+1} - x^n}{\tau_{n+1}} + Ax^n = f, \tag{13}$$

where B is a square nonsingular matrix of the n th order, it is called the preconditioner, τ_{n+1} is a number, which is called the iterative parameter.

The preconditioner B of the alternately triangular method can be written as

$$B = (D + \omega R_1)D^{-1}(D + \omega R_2), D = D^* > 0, \omega > 0, y \in H,$$

where R_1, R_2 are the lower and upper triangular operators (matrices) $A_0 = R_1 + R_2, R_1 = R_2^*$, where D is some operator (for example, the diagonal part of the operator A).

5 Estimation of acceleration of parallel algorithms for solving grid equations by iterative methods based on MPI

Fig. 4 shows a comparison of the acceleration of the parallel algorithm depending on the amount of transmitted data on a grid of 10,000 x 10,000 computational nodes using the Seidel method. Measurements of the calculation time were made for transmissions with a volume of 5, 10, 50, 100, 500, and 1000 elements. The greatest acceleration was observed with a transmission volume of 100 elements. With an increase in the volume of receiving and transmitting data, the speed of calculations began to decrease. This result is due to the fact that with large volumes of transfers, labor costs for exchanges between computing nodes increase, which ultimately does not justify itself.

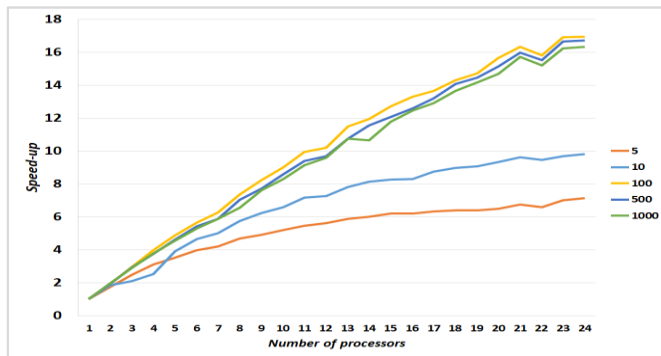


Fig. 4. Comparison of acceleration depending on the amount of data transferred.

The time spent on the parallel implementation of one iteration of the Seidel method is

$$t(m) = 9t_a(N_x - 2 + m(R - 1))(N_2^r - 2) + 2 \sum_{s=0}^{R-2} t_l(s) + (Q - R + 1) \cdot t_l(R - 1) + 2t_x(N_x - 2). \tag{14}$$

where m is the amount of transmitted data, Q is the number of blocks, s is the step number, $t_l(s)$ is the time of organization of data transfer (latency).

Take the derivative $t(m)$ with respect m to $Q = \left\lceil \frac{N_x - 2}{m} \right\rceil \approx \frac{N_x - 2}{m}, N_2^r \approx N_2.$

$$t'(m) = 9t_a(R-1)(N_2-2) - \frac{N_x-2}{m^2} \cdot t_l(R-1) = 0. \quad (15)$$

Therefore, the optimal transmission volume is

$$m = \sqrt{\frac{(N_x-2) \cdot t_l(R-1)}{9t_a(R-1)(N_2-2)}}. \quad (16)$$

Similarly, the optimal transmission burst size for the modified alternating-triangular method [9] is calculated.

Fig. 5 shows a comparison of the theoretical and practical values of acceleration in the case of the optimal amount of gears.

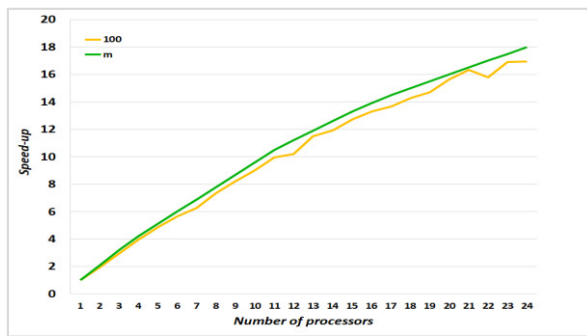


Fig. 5. Comparison of theoretical and practical acceleration values in the case of optimal transmission volume

Fig. 6 shows a comparison of the acceleration for the Seidel methods with the optimal amount of transmitted data and Jacobi depending on the number of computing nodes. The calculations were made on a grid of 1000 by 1000 cells. The launches were carried out sequentially, starting from the launch on one computing node and ending with the connection of all available nodes.

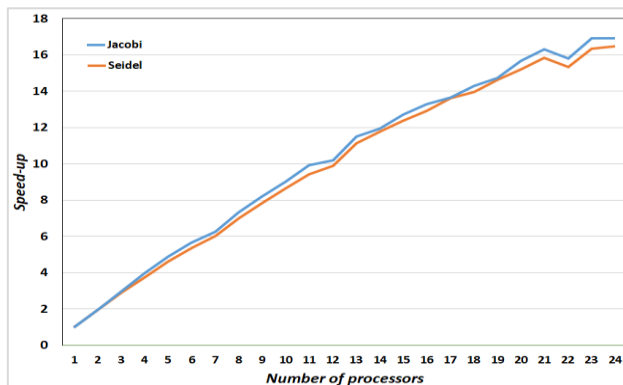


Fig. 6. Variability of acceleration of Seidel and Jacobi methods as a function of the number of computational nodes

It can be seen from the figure that the acceleration of the algorithm that implements the Seidel method is not significantly inferior to the acceleration of the algorithm for the Jacobi method.

6 Conclusion

In the course of this work, a software package was developed that makes it possible to perform calculations for the problem of the transfer of matter in a shallow reservoir on various computational grids. Theoretical estimates are made to find the latency time. The parallel algorithms implemented in the software package are oriented on a multiprocessor computer system K-60 in KIAM RAS. They can significantly reduce the time of the software package with a large amount of input data. A number of experiments were carried out with different amounts of transfers for a varying number of computing nodes. Optimal volumes of transmitted data have been obtained. The presented complex can be used to study the processes of pollutant transfer in natural and technological systems.

Acknowledgments

The study was supported by the Russian Science Foundation grant No. 21-71-20050.

References

1. E. Alekseenko, B. Thouvenin, J. Tronczynsky, F. Carlotti, P. Garreau, C. Tixier and M. Baklouti, *Marine Pollution Bulletin*, **128**,140–155 (2018). DOI: 10.1016/j.marpolbul.2018.01.008 (2018)
2. V. Badenko, *High-performance computing* (St. Petersburg: Publishing House of the Polytechnic University, 2010)
3. A. Sukhinov, A. Chistyakov, A. Shishenya and E. Timofeeva, *Predictive Modeling of Coastal Hydrophysical Processes in Multiple-Processor Systems Based on Explicit Schemes Mathematical Models and Computer Simulations*, **10(5)**, 648-658 (2018). DOI 10.1134/S2070048218050125 (2018)
4. A. Atayan, V. Dolgov, *Journal of Physics: Conference Series*, **2131 (2)**, DOI: 10.1088/1742-6596/2131/2/022008 (2021)
5. C. Douglas, Montgomery, Elizabeth A. Peck, G. Geoffrey Vining *Introduction to Linear Regression Analysis*. 6th edn. Wiley. USA (2012).
6. D.G.Kleinbaum, L.L.Kupper, A. Nizam, E.S. Rosenberg, *Applied Regression Analysis and Other Multivariable Methods*. 4th edn. Cengage Learning, USA (2013).
7. Jr. Harrell, E. Frank, *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. 2nd ed. Springer Series in Statistics. Switzerland (2015).
8. M. Kuhn, K. Johnson *Applied Predictive Modeling* (2nd ed. Springer. Switzerland, 2018).
9. A.A. Samarsky, A.V. Gulin, *Numerical methods: Textbook, manual for universities*, (Nauka, M., USSR, 1989).