Specialized Web Portal for Solving Problems on Multiprocessor Computing Systems

Elena N. Akimova[1,2], Vladimir E. Misilov[1,2],
Aliya F. Skurydina[1,2], and Maxim P. Martyshko[3]*

[1] Ural Federal University, Yekaterinburg, Russia,
[2] IMM UB RAS, Yekaterinburg, Russia,
[3] IG UB RAS, Yekaterinburg, Russia
aen15@yandex.ru,{out.mrscreg,infofreeru,afinapal}@gmail.com

Abstract. A system for remote calculations called "Specialized web portal for solving problems on multiprocessor computing systems" has been developed and installed at the Department of Ill-Posed Problems of Analysis and Applications of the Institute of Mathematics and Mechanics UrB RAS. The parallel algorithms have been incorporated into this system to solve the inverse gravity problem of lateral density reconstruction, the structural inverse gravity and magnetic problem of the contact surfaces reconstruction, and solving SLAEs with block-tridiagonal matrices in geoelectrics problems.

Keywords: parallel algorithms · inverse and ill-posed problems · iterative algorithms · web development

1 Introduction

The modern physics widely uses mathematical modeling for solving different problems. The solution of some problems is very complex process which requires the cutting-edge mathematics, as well as the highest level of computerization. Firstly, the solution of problems requires storage and processing of large amounts of data. Secondly, the algorithms have high complexity, *i. e.* increasing the input data size substantially increases the computation time. Solving this problems using a personal computer takes from several hours to several days. In this case the parallel computing can be a great aid. Using the multiprocessor systems or GPU's reduces the calculation times by several orders of magnitude.

On the other hand, the researcher has to face many challenges. He must be an expert in his subject area, and also must be a mathematician to use algorithms for solving complex equations. Implementation of these algorithms on modern parallel computing systems requires not only good programming skills but also knowledge of special technologies such as MPI or CUDA.

* This work was partly supported by the Ural Branch of the Russian Academy of Sciences Fundamental Research Programs, projects 15-7-1-3 & 15-7-1-13. The work was partly supported by Act 211 Government of the Russian Federation, contract 02.A03.21.0006.

Apparently, getting scientific results is a complex and time-consuming process which requires expensive human resources and powerful computing systems.

Thus, there is a need for a specialized web portal which would greatly facilitate and reduce the cost of these processes. There are several solutions like [1], but they are devoted mostly to the technical issues and not to the mathematical algorithms.

2 General description

The web portal [2, 3] via the web interface allows the user to choose the required remote computing system type, specify the number of processor nodes or GPUs, the problem type, and the method of its solution, load the input data, obtain the output data, and display the solution results using the Surfer and gnuplot graphic packages. The calculation time for each task is also shown.

The web portal consists of three major parts shown on Fig. 1: IIS HTTP server with a web application; MS SQL Server 2008 to store all user tasks with input and output data; a service to perform the data transfer, to launch tasks on various computers, to determine task status, and to transfer the results into the web portal.

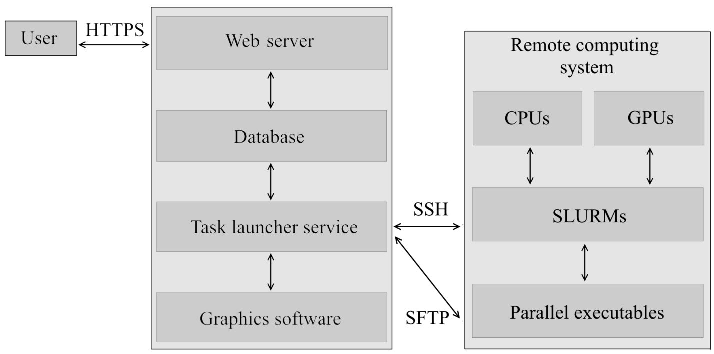

Fig. 1. Web portal structure.

2.1 Web application and database

Web application is developed using C# and ASP.NET technology.

The user has to be registered in the system to create or to view tasks. To create a new task the user must choose a required task from the prepared list of available task types. Some of the parameters are defined by the task type, such as grid size, desired accuracy and boundary conditions while the others are defined by the computing system type, such as cores and nodes numbers and execution time limit. To launch a task the user must specify all required parameters, choose a computer type and upload the input files. The task data are stored in the database.

After launching his task the user can watch the progress in a separate page. The task is considered successfully completed when the program on a computing system finishes without errors and all output files are transferred to web server. The page for completed task shows the actual execution time and allows user to download the output files (Fig. 2). It also demonstrates images of output file created using Golden Software Surfer & Voxler or gnuplot graphics software.

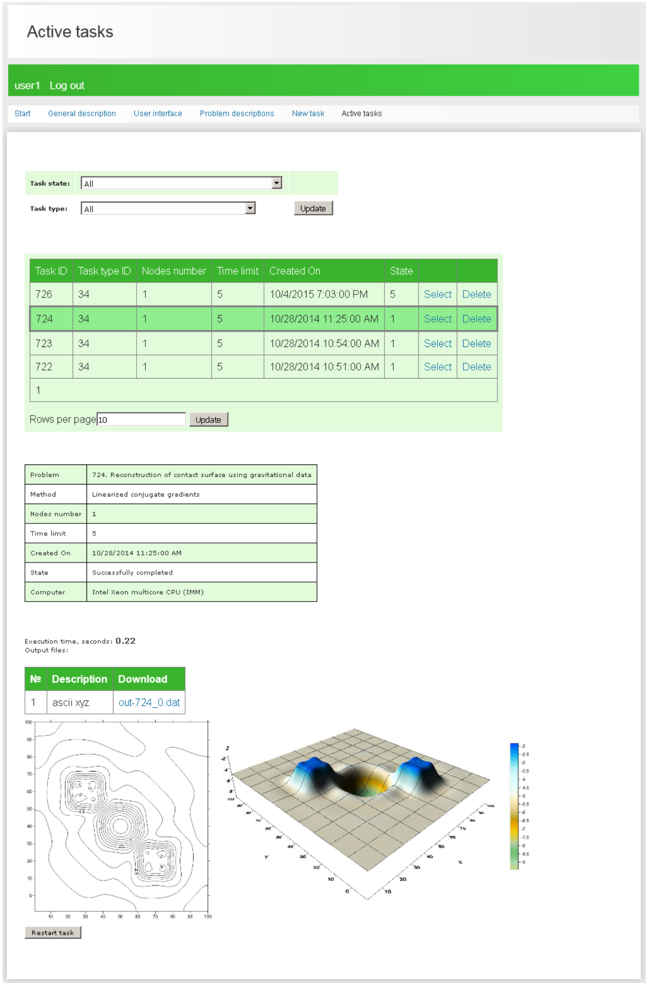

Fig. 2. Active tasks page and finished task

2.2 Task launcher service

The algorithm includes the following steps:

1. Look for a new task in database.
   1.1. Read the task input parameters.
   1.2. Validate the input files for proper format and structure.
   1.3. Transfer the input files to the computing system.
   1.4. Launch the task solving executable on the computing system.
   1.5. Store the launched program id in the database.
2. Look for a launched task in database.
   2.1. Check for a finished program on the computing system.
   2.2. Transfer the output files to the web server.
   2.3. Create an image of output files using graphics software.
   2.4. Mark the task as finished.

The service uses SSH and SFTP protocols for remote interaction and file transfer. Currently the service supports interaction with remote computing systems with the Slurm Workload Manager. This system is widely used by supercomputers and computer clusters and it provides a framework for executing and monitoring programs for CPUs and GPUs. The service generates a batch script which transfers input data to the executable and launches it using the **sbatch** command. This command returns the task ID which is used to check for a finished program utilizing the **sacct** command.

After retrieving the output data from the remote system the service transfers it to the database and to graphics software. It launches a prepared script for Surfer or gnuplot which plots a preview of output files according to the task type. For example, for the structural inverse gravity problem of finding a contact surface, the result would be a 3D surface plot created by Surfer.

3 Available task types

Currently, the specialized web portal can be used to perform tasks for solving gravity problems (finding a lateral density, reconstruction the contact surfaces), magnetic problems (reconstruction the contact surfaces), field separation by depth problem and solve SLAEs with block-tridiagonal [4] and block-fivediagonal [5] matrices for geoelectrics problems using the original parallel algorithms developed by authors.

3.1 Field separation

The problem of separating gravity or magnetic fields by depth is solved using the technique by P. S. Martyshko & I. L. Pritkin [6]

This technique consists of several steps. The first is to exclude side sources of field. To do this we must find the solution of Dirichlet problem in the considered area and subtract it from the original field. The portal uses the parallel algorithm based on the separation of variables method [7]. The next step includes solving the integral equation based on the Poisson integral for a half-space. This problem is reduced to SLAE and solved by the minimal error method [8, 2, 3].

3.2 Finding a lateral density

The problem of finding a variable density in a horizontal layer is described by a linear two-dimensional Fredholm integral equation of the first kind:

$$A\sigma = f \int_{\infty}^{\infty}\int_{\infty}^{\infty} \left\{ \frac{1}{\sqrt{(x-x')^2+(y-y')^2+H_1^2}} - \frac{1}{\sqrt{(x-x')^2+(y-y')^2+H_2^2}} \right\} \sigma(x',y')dx'dy' = \Delta g(x,y),$$

where f is the gravitational constant, $\Delta g(x,y)$ is the known gravitational field, and $\sigma(x',y')$ is the desired density distribution function.

After the discretization into $n = M \times N$ grid of the area $D = \{(x,y) : a \leqslant x \leqslant b, c \leqslant y \leqslant d\}$ and approximation of the integral operator A using quadrature rules, it takes the form

$$A\sigma = f \sum_{i=1}^{M}\sum_{j=1}^{M} \left\{ \frac{1}{\sqrt{(x_v-x_i)^2+(y_u-y_j)^2+H_1^2}} - \frac{1}{\sqrt{(x_v-x_i)^2+(y_u-y_j)^2+H_2^2}} \right\} \sigma_{(i-1)M+j}\Delta x \Delta y = b_{(v-1)M+u}, \quad u = \overline{1,M}, v = \overline{1,N}. \tag{1}$$

Equation (1) is a system of linear algebraic equations with a symmetric positive defined matrix of $n \times n = MN \times MN$ dimension

$$\bar{A}z = b.$$

This SLAE is solved by the parallel minimal error method [8, 2, 3].

3.3 Structural problems

The gravity and magnetic problems of reconstruction of the contact surfaces are described by the nonolinear two-dimensional Fredholm integral equations of the first kind:

$$A(\zeta) \equiv f\Delta\sigma \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left\{ \frac{1}{\sqrt{(x-x')^2+(y-y')^2+\zeta^2(x,y)}} - - \frac{1}{\sqrt{(x-x')^2+(y-y')^2+H^2}} \right\} dxdy = \Delta g(x',y',0),$$

$$B(\zeta) \equiv \Delta J \int\limits_{-\infty}^{\infty} \int\limits_{-\infty}^{\infty} \left\{ \frac{\zeta(x,y)}{[(x-x')^2+(y-y')^2+\zeta^2(x,y)]^{3/2}} - \right.$$
$$\left. - \frac{H_l}{[(x-x')^2+(y-y')^2+H_l^2]^{3/2}} \right\} dxdy = \Delta Z(x',y',0).$$

where f is the gravitational constand, $\Delta g(x',y',0)$ is the known gravitational field, $\Delta Z(x',y',0)$ is the known magnetic field, and $\zeta(x,y)$ is the function which describes the desired surface.

After the discretization into $n = M \times N$ grid of the area $D = \{(x,y) : a \leqslant x \leqslant b, c \leqslant y \leqslant d\}$ and approximation of the integral operators A or B using quadrature rules, this equations take the form:

$$\bar{A}[z] = \bar{F}.$$

This equation is solved by the componentwize Newton method [9, 10] or the linearized conjugate gradients method [11].

4 Conclusion

Each algorithm has to be researched and tested on different types of data before implementing to the system.

Now the authors are developing modules for solving problems of finding several contact surfaces simultaneously by the total gravitational field and magnetic problem of finding the contact surface in the case of an arbitrary magnetization direction.

The problem of constructing the schedule for trains on the single-track railway will be incorporated soon.

The test version of the web portal currently works with the Uran supercomputer of the Institute of Mathematics and Mechanics. It allows to run task on the multicore Intel Xeon processors or NVIDIA Tesla GPUs.

References

1. Sukhoroslov O., Afanasiev A. Everest: A Cloud Platform for Computational Web Services. SCITEPRESS, Proceedings of CLOSER 2014, pp. 411–416.
2. Akimova, E. N. and Gemaidinov, D. V., Parallel Algorithms for Solving the Inverse Gravimetry Problem and Organization of Remote Communication between PCS1000 and the User (in Russian), Numerical Methods and Programming, 2008, vol. 9, no. 1, pp. 133–144.
3. Akimova, E. N., Belousov, D. V., and Misilov, V. E. Algorithms for solving inverse geophysical problems on parallel computing systems. Numerical Analysis and Applications, 2013, vol. 6, no.2, pp. 98–110.
4. Akimova E. N., Belousov D. V., Parallel algorithms for solving linear systems with block-tridiagonal matri. ces on multi-core CPU with GPU. Jornal of Computational Science, 2012, Vol. 3, Issue 6, pp. 445-449.

5. Akimova E. N., A parallel matrix sweep algorithm for solving linear system with block-fivediagonal matrices. Proc. of ICNAAM 2014. AIP Conference Proceedings, 2015, Vol. 1648, UNSP 850028.
6. Martyshko P. S., Prutkin I. L., Technology of depth distribution of gravitational field sources (in Russian). Geophys. J, 2003, vol. 25, no. 3, pp. 159–168.
7. Martyshko P. S., Fedorova N. V., Akimova E. N., Gemaidinov D. V., Studying the Structural Features of the Lithospheric Magnetic and Gravity Fields with the Use of Parallel Algorithms. Izvestiya, Physics of the Solid Earth, 2014, vol. 50, no. 4, pp. 508–513.
8. Vasin V. V.and Eremin I. I. Operators and iterative processes of Fejer type: theory and applications. Walter de Gruyter, 2009, vol. 53.
9. Akimova, E. N., Misilov, V. E., and Skurydina, A. F. Parallel algorithms for solving a structural inverse mag-netic problem on multiprocessing computer systems (in Russian). Vestnik UGATU, 2014, vol. 18, no. 4(65), pp. 206–215.
10. Akimova E. and Skurydina A. A componentwise Newton type method for solving the structural inverse gravity problem. Proceedings of 14th EAGE International Conference on Geoinformatics - Theoretical and Applied Aspects, 2015.
11. Akimova, E. N., Misilov, V. E., Skurydina, A. F., and Tretyakov, A. I. Gradient methods for solving inverse gravimetry and magnetometry problems on the Uran supercomputer (in Russian). Numerical Methods and Programming, 2015, vol. 16. pp. 155–164.