RUSSIAN JOURNAL OF EARTH SCIENCES
vol. 13, 1, 2013, doi: 10.2205/2013ES000528

Implementation of seismic data inversion as grid-backed web service

Kirill I. Kholodkov

Schmidt Institute of Physics of the Earth RAS, Moscow, Russia

Abstract. The essence of probability approach in solving inverse problems is use of a posteriory probability density function (APDF). APDF bears a priory system signature and the way the corresponding model parameter set describes the observational data. APDF importance sampling is carried out independently in different points thus the problem is a good target for distributed computing. The software implementation make use of virtualization to lower the hardware cost: all Globus Toolkit services, except for GridFTP, run as virtual guests on execution nodes. Due to very insignificant resources utilization the guests make no footprint on node's computation power. The execution nodes local resource manager is TORQUE. The main thought for web interface is to hide complex middleware interaction from user. Such interaction usually takes place during start-up and result collection

This is the e-book version of the article, published in the Russian Journal of Earth Sciences (doi:10.2205/2013ES000528). It is generated from the original source file using LaTeX's **epub.cls** class.

and analysis procedures. The implemented minimalistic web service allows defining of models, estimation of calculation time for user-specified parameter set, task start-up, management, results acquisition and visualization. The specific pilot problem we implemented is determination of seismic anisotropy of mantle and upper crust by teleseismic data.

1. Introduction

This project introduces a grid web-service designed to solve the specific class of inverse problems in geophysics. The main idea behind the solution of this specific inverse problem is exploiting an a posteriori probability distribution function (APDF) which reduces the problem to tabulation of a function of several variables. Since the APDF value is computed in each point independently the distributed computing systems fits computation process perfectly [*Aleshin et al.*, 2009]. Distributed computing systems are supported by special software called the middleware, which primary goal is to organize multiuser workflow in distributed environment while securing the system against malicious use. Distributed systems running such software are called grids. The middleware falls into two groups by kind of the computing resources it uses: idle processing power of personal computers (BOINC, XtremWeb, etc) and power of dedicated computers (Globus Toolkit, UNICORE, ARC, etc). Grids of the first group are often called volunteer grids. Ones consisting of dedicated high performance computers are using the second middleware group, which is represented by Globus Toolkit, gLite, UNICORE, ARC, etc. Most implementations of middleware include five base components. These are user authentication, task execution and/or programming interfaces, task scheduler, data delivery (stage-in and stage-out in terms of grid) and monitoring services. The abovementioned components may vary on functionality and complexity, be part of the third-party software or even be absent depending on hardware infrastructure design, type of solved tasks, or access policy. Most middleware have generic user authentication and authorization architecture, which utilizes the X.509-based Public Key Infrastructure, PKI (Internet X.509 Public Key Infrastructure: Certificate and CRL Profile, http://www.ietf.org/rfc/rfc2459.txt). Such architecture impose upon the following: before the user could access the grid a certificate from corresponding certification authority must be obtained, after that in order to submit a task the user has to acquire a proxy-certificate which is later used by every grid. To start a task on grid one should use middleware-specific mechanisms. Grids based on Globus Toolkit or gLite use Job Description Language (JDL) to allow specification of execution parameters, data delivery and other actions. JDL-like way is often under-documented and its syntax tends to change rapidly throughout version. Thus the grid is not an easy to use instrument because user is required to have specific knowledge and skills. There is a set of software solutions that address the complexity of user to grid interaction called grid application frameworks [*Allen et al.*, 2005]. Such frameworks provide simplified cross-platform programming interfaces with grid, allowing easy access to grid infrastructure for application developers [*Berman et al.* 2001; *Fahringer et al.* 2005]. However still many middleware-specific features are not hidden from user. For testing purposes a model from previous work *Aleshin et al.*, 2009 is adopted, as shown in Table 1.

Another approach implies the creation of interface to grid, allowing users to solve a single class of problems, for example, a collection of web-services for data source access, compute applications, visualization applications and mapping, the QuakeSim (`http://www.`

quakesim.org/). The inability to be user-modified or be installed on users site are disadvantage of particular product but such solutions are the most content way of interaction with grids from users perspective. We have developed a custom grid application in the same way as mentioned above hiding most of technical issues of grid computing. The application interacts with user via web-browser, which makes use of complex distributed systems affordable for the most of users. Presented grid-application consists of web-interface and the middleware connector. Web-interface provides additional security besides being easy to use by restricting the programs that could be potentially used within grid. The latter allows users not to authenticate to grid middleware and evade the majority of difficulties associated with this process. The middleware connector is a set of programs and scripts which enables creation of packages with configuration data coming from web-interface and other program components and submitting the package to execution by means of grid middleware. Additionally the connector enables user to monitor tasks and also gathers computing results.

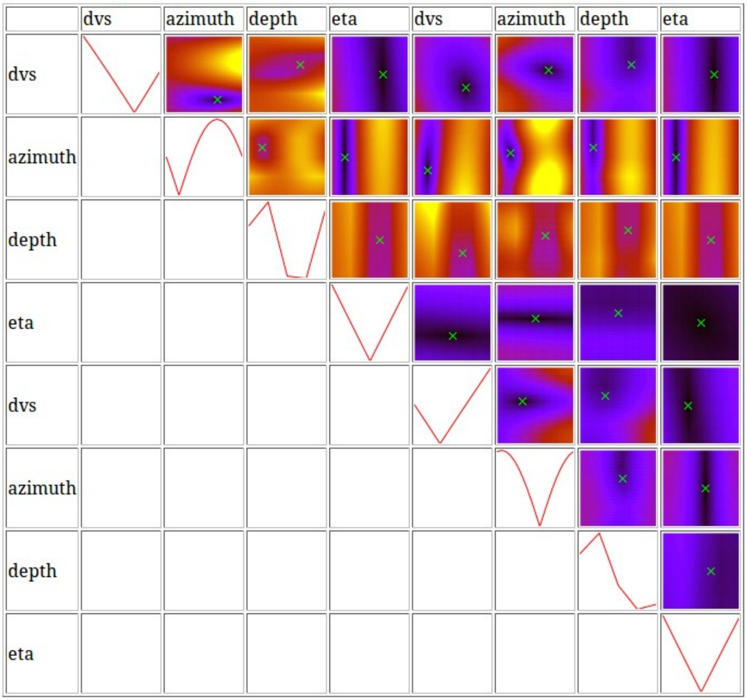

Figure 1. Sections of objective function (single- and two-dimensional) at its absolute minimum. The axes of each graph is defined in corresponding left column and top row. Cross depicts the minimum.

2. The Problem

In this paper the Earth crust and upper mantle seismic anisotropy determination from converted waves problem is used as an example (see Figure 1). Details of data processing and model parameterization are stated elsewhere [*Vinnik et al.*, 2007]. The objective function is root-mean-square deviation of observational seismic data and synthetic seismic data. To compute the latter a layered halfspace model is used. The problem was already ported to run on grid. For testing purposes a model from previous work [*Aleshin et al.*, 2009] is adopted (see Table 1).

Receiver function waveforms were computed for the abovementioned model, those were later used as an observational data. The optimization now includes the anisotropic parameters η for each layer. The computation was performed on the rectangular mesh with the following parameters: S-wave anisotropy coefficient varies from 0 to 0.05 with step 0.01, η varies from 1.00 to 1.05 with step 0.01. Layer thickness parameter reside within 50 to 100 kilometers with 10 kilometer step, anisotropy axis azimuth varies from 0 to 175 degrees with 5°step.

Table 1. Example model adopted from [*Aleshin et al.*, 2009] with column η which is added in this project

Layer number	V_p, km s^{-1}	V_s, km s^{-1}	ρ, g cm^{-3}	Thick, H, km	Azimuth, ϕ, deg.	Anisotropy coefficient α	η
1	6.4	3.7	2.9	40	–	0.00	1.00
2	8.1	4.5	3.3	60	30	0.04	1.03
3	8.1	4.5	3.3	100	100	0.02	1.03
4	8.5	4.72	3.4	∞	–	0.00	1.00

3. Web Interface and Middleware Connector

The project makes use of custom-built web-interface to interact with user. This interface has three components: execution parameters setup, task submission with monitoring and result visualization. The client-server way of user interaction allows most of the functionality to be implemented at server side with common web technology: PERL and PHP. To perform a computation one should set a number of parameters on the webpage. The interface for setting the parameters is a dynamic webpage and contains a number of text fields. Gathered statistics enabled user to perform runtime estimation with defined parameters, however this feature is still under development. One of the notable components of grid application is task package compiler. Each package (which is actually a compressed tar-ball) consists of program binary, source data, configuration data and a set of maintenance scripts. This package is submitted to middleware internals for execution on compute elements.

When the task has been set up and the user places it for execution, the middleware connector gathers user-defined parameters and program components into a single package and submits it by means of middleware for execution. At this stage the user can monitor the task processing and execution. During the computation for the purpose of storage, transmission and post processing convenience the results are held in a relational database. The values of multidimensional function cannot be sorted with simple indexing because result analysis implies cross-sectioning within parameter space in different directions. This leads to adoption of multidimensional indexing mechanism in order to enable online data access, therefore a supported database management system (DBMS) is required, MySQL in this case. It should be noted that multidimensional indexing takes considerable time of several hours and does not benefit from being executed in high-performance environment. The user can download the result database for further analysis. However the web-interface provides the means of initial result preview (Figure 1), which is implemented as a set of graphs with single-dimensional and two-dimensional cross-sections of the objective function at its absolute minimum.

Besides the larger size, images on Figure 2 and Figure 3 also contain detailed information of values, initial and final values with a color legend for 2-dimensional section.

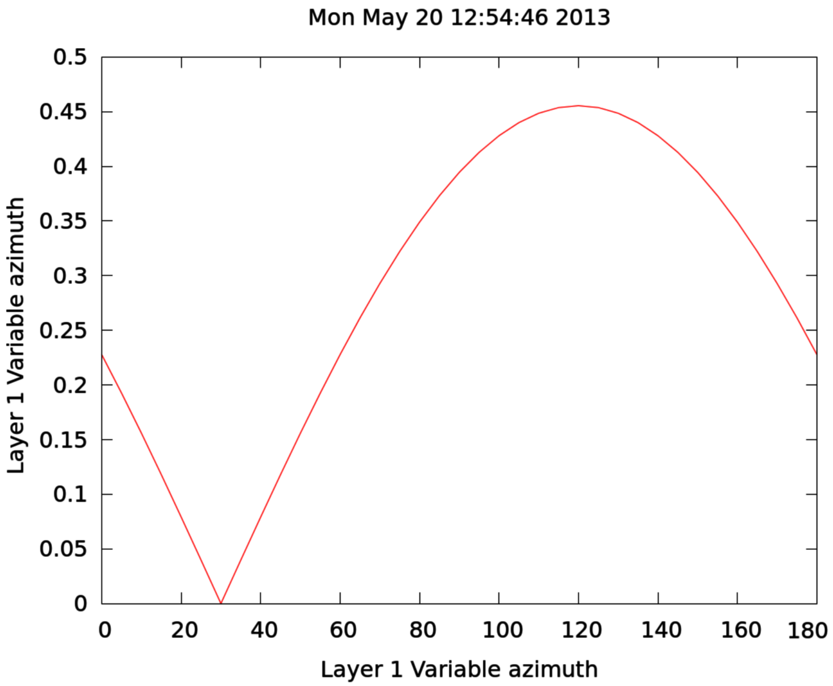

Figure 2. Enlarged images of 1D graph from previous figure.

4. Grid Infrastructure

Computations were performed on grid site specially deployed for this project. The project makes use of Intitute of Physics of the Earth of Russian Academy of Sciences (IPE RAS), All-Union Institute for Scientific and Technical Information RAS and Geophysical Center

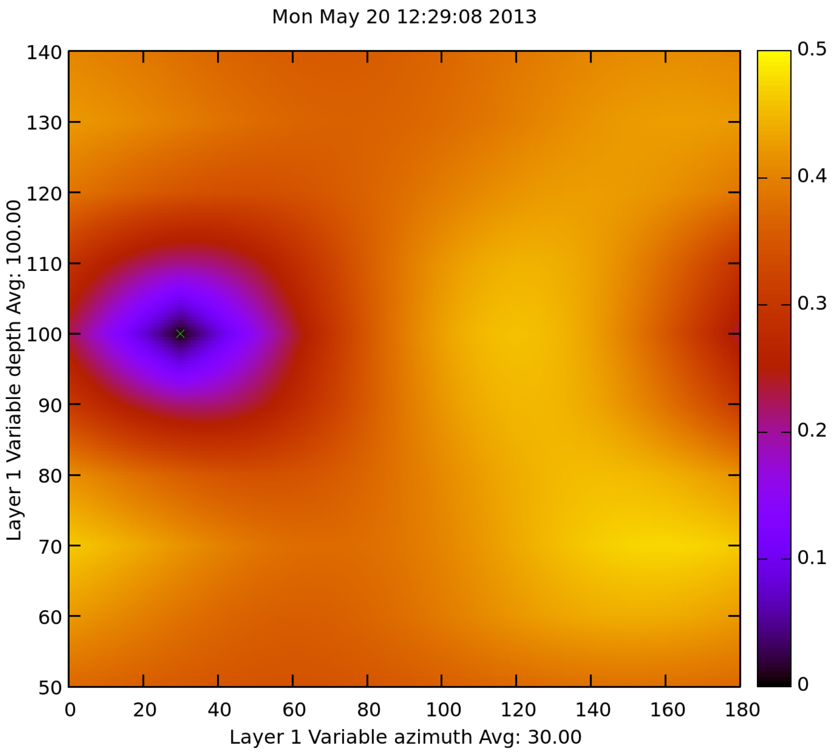

Figure 3. Enlarged images of 2D graph from previous figure with colorbar.

of Russian Academy of Sciences (GC RAS) hardware. The IPE cluster is a blade system based on Intel Modular Server technology. Each compute module has two Intel Xeon E5520 processors and 12 gigabytes of RAM. Node interconnection is gigabit Ethernet. Storage systems allows simultaneous access to data. The VINITI cluster is five standard servers with 2 E5430 Xeon processors each. Shared storage isn't implemented in hardware and is provided via software. GC RAS provided a server with 1 Tb of redundant storage. This server is not used as compute element rather than as a storage. Deployed grid is based on Globus Toolkit middleware. The components of the middleware used within this project are: certificate authority, identity storage, file storage, local resource manager and local task scheduler. Certificate authority is a toolset supporting Public Key Infrastructure that provides authentication and authorization across all grid services. This project utilizes the OpenSSL toolset. The project no longer requires user to interact with grid security. All computation is done as single special user. This approach required additional software for grid application to grid middleware communication, because Globus Toolkit based grids use proxy-certificates instead of user certificates to authenticate within grid. Proxy certificate is generally created by user however grid application has its own mechanisms and dedicated identity. Proxy certificate creation occurs in two steps: new certificate is generated with new key pair which is then signed with users key. Proxy certificate is very short-living and could be used by grid software on behalf of user. Centralized certificate repository allows some automation for proxy certificate generation and is provided by MyProxy service. Data storage and access is supported with GridFTP, an FTP extension with improved access control and many other significant improvements. GridFTP security is based on Globus Security Interface which is PKI-based. Local resource manager starts and stops computational tasks within the cluster. This software is generally specific to each cluster however many middleware products contain adapters for most popular resource managers. This project uses TORQUE on both clusters. Besides the local resource manager clusters use local resource scheduler, which are mostly a part of local resource manager but can be replaced with often commercial dedicated software. All grid-infrastructure services except for GridFTP, and also web interface and middleware connector reside on virtual machines. Virtualization allows independent instances of operating systems to be launched on the same hardware as the host operating system. Hardware acceleration for virtualization eliminates the overhead and greatly reduces the hardware cost. Subsequently the project would have a virtual private network that will allow virtual machines to migrate to different cluster and maintain operability while its former host is down for maintenance, for example.

5. Conclusion

The use of described grid application to solve modeling problems made computation with distributed system significantly easier. However the software implementation is far from being optimal and requires improvement. Middleware connector could be improved to be more lightweight while the computational task could be more sophisticated.

Acknowledgments. I expresses gratitude to I. M. Aleshin (IPE RAS) for problem formulation and attendance. The work is supported by RFBR grants 10-07-00491-a, 11-05-00988-a and 11-07-12045-ofi-m-2011.

References

Aleshin, I. M., M. N. Zhizhin, V. N. Koryagin, D. P. Medvedev, D. Yu. Mishin, D. V. Peregoudov, and K. I. Kholodkov (2009), Use of distributed computing systems in seismic wave form inversion, *Russ. J. Earth Sci., 11*, doi:10.2205/2009ES000415.

Allen, G., et al. (2005), The grid application toolkit: Towards generic and easy application programming interfaces for the Grid, *Proceedings of the IEEE, 93(3)*, 534–550, doi:10.1109/JPROC.2004.842755.

Berman, A., et al. (2001), The GrADS Project: Software support for high-level grid application development, *International Journal of High Performance Computing Applications, 15*, 327–344.

Fahringer, Thomas, et al. (2005), ASKALON: A Grid Application Development and Computing Environment, *GRID'05. Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, 122–131.

Vinnik, L. P., I. M. Aleshin, S. G. Kiselev, G. L. Kosarev, and L. I. Makeyeva (2007), Depth localized azimuthal anisotropy from SKS and P receiver functions: The Tien Shan, *Geophysical Journal International, 169*, 1289–1299, doi:10.1111/j.1365-246X.2007.03394.x.