Distant Computation of Travel-time Sensitivity Kernels over the Internet

Y. Saidi[1,*], L. Gizon[1], A. C. Birch[2], J. Jackiewicz[1]

[1] Max Planck Institute for Solar System Research (MPS), 37191 Katlenburg-Lindau, Germany.

[2] Colorado Research Associates (CoRA), a division of NWRA Inc., Boulder, CO 80301, U.S.A.

[*] Supported by European Helio- and Asteroseismology Network (HELAS).

Abstract

We describe a mechanism based on open-source software technologies, which allows local and distant users to easily run computations at the Max Planck Institute for Solar System Research (MPS) over the World Wide Web (WWW). This system was developed to run MATLAB, C and FORTRAN codes through a Web browser interface. In this poster we discuss the implementation of MATLAB codes to compute travel-time sensitivity kernels for local helioseismology. Multiple computation jobs can be distributed over several machines without the need for a specific manipulation or additional MATLAB licenses, and the user is notified of the status of the computation by email.

Introduction

Rapid advances in WWW and networking technologies have made it possible to design computation systems that integrate resources located at distant locations.

The main goal of this work is to take advantage of these advances to make helioseismology software and computer resources accessible to the scientific community, in a completely transparent manner.

Here, we want to make available two different codes over the internet to compute travel-time sensitivity kernels: 3D sound-speed kernels (Birch et al. 2004, see Fig. 1) and 2D flow kernels (Jackiewicz et al. 2007, see Fig. 2). Both codes are written in MATLAB.

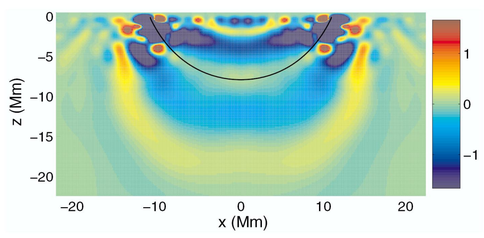


Fig. 1. *Example of a cut through a 3D kernel giving the sensitivity of travel times to sound speed perturbations* $\delta c^2/c^3$. *The distance between the two points at the surface is* 20 Mm *and the units of the kernel are* 10^{-2} Mm *(Birch et al. 2004).*

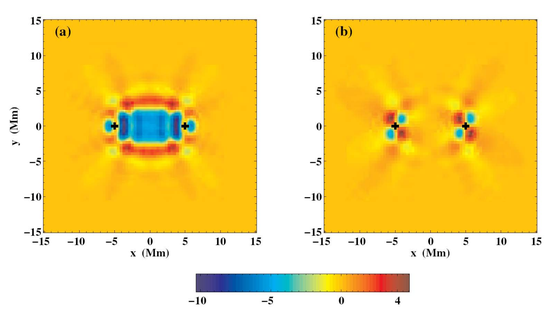


Fig. 2. *Example of an f-mode 2D kernel giving the sensitivity of travel times to flows. The distance between the two points is* 10 Mm *and The units of the kernel are* s Mm^{-2} $(km/s)^{-1}$ *(Jackiewicz et al. 2007).*

In general, kernel codes have a lot of parameters as input values. An example input file to Birch's code is shown in Fig. 3. An important aspect of this project is to design a web interface that will make it easier for external users to use all of the kernel code capabilities.

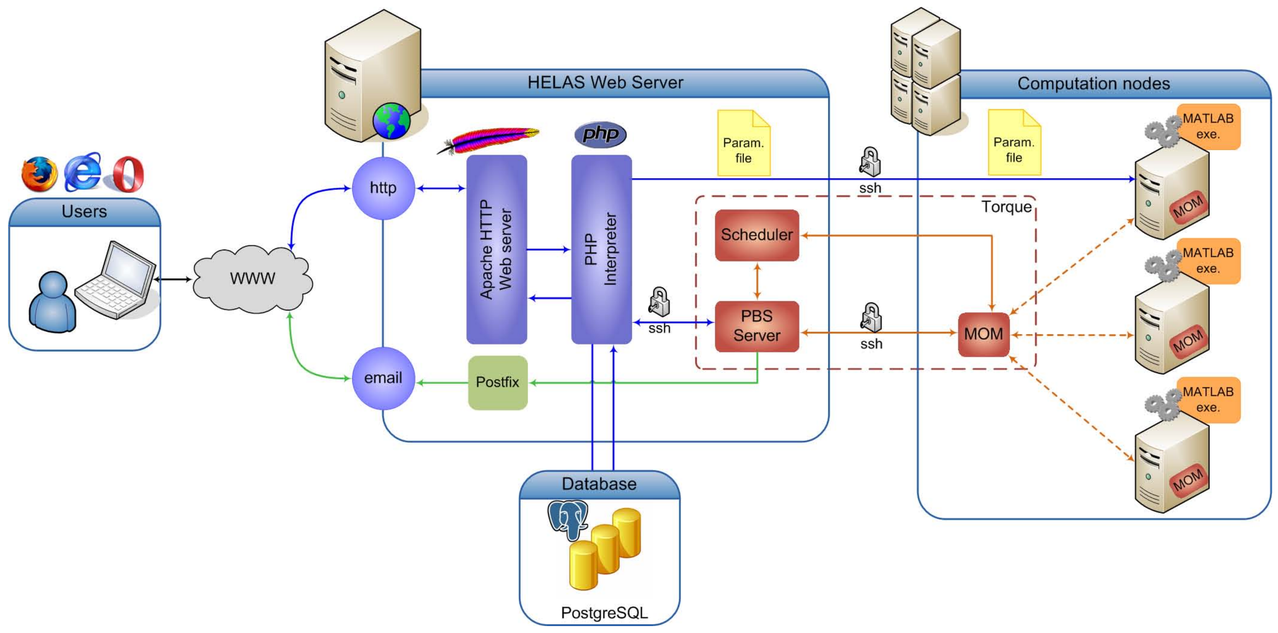
% output directory % /home/aaronb/data2/fmode_sound_speed/ % % which calculation to do: % power % hires_power % zero_order % TD_sound % TD_flow % td_sound % % holography green's % none % % pupil % none % % solar model file % /home/aaronb/projects/kc/model % % filter file % /home/aaronb/projects/kc/fmode_only_filter % % which OTF % 0 for no otf % 1 % % minimum distance in Mm % 5 % maximum distance in Mm % 15 % % number of distances to compute % 3	% % guess times % fmode_window % % greens functions type % normal_mode % % eigenfunction files % /home/aaronb/data2/eigs_mr_damp2/model_eig % % damping file % damping2 % % maximum value of n % 4 % % max mode frquency (mHz) % 600 % % source depth (km) % 100 % % source type % 1==vertical velocity % 2==divergence % 3==quadrapole % 3 % % observation depth (km) % -200 % % correlation time (seconds) % 48	% % min frequency (mHz) % 1.5 % % max frequency (mHz) % 4.5 % % number grid points in frequency % 200 % min k (Mm^-1) % 0.2 % % max k (Mm^-1) % 1.8 % % number grid points in k % 1000 % % number of gridpoints in x direction % 300 % % number of gridpoint in y direction % 100 % % grid in acoustic depth % depth_grid % % quiet mode % yes or no yes

Fig. 3. *Typical input parameter text file to compute travel-time sensitivity kernels for sound speed perturbations.*

As we designed the distant computing system we were faced with a number of complications which did not have obvious solutions. MATLAB is a useful tool as an analysis development platform, but MATLAB is not easy to integrate as part of a wider system (multi-user, distributed computing, user licences). There exists a commercial MATLAB system for distributed computing, which is, however, very expensive.

Instead, we chose to design a system based on open-source technologies. This solution requires some work to learn how to interface the various components of the system, as explained in the next section.

Fig. 4. *Global architecture of the system for distant computing. The HELAS Web server houses the Apache Web server, the PHP interpreter, and the main part of the Torque job management system.*



Solution and architecture

A Web-based approach was chosen because Web browsers are the easiest, platform-independent way of communication between the remote user and the local computation facilities.

Our system architecture (see Fig. 4) relies on the following components:

- **Apache web server.** Apache is used to make a web page available over the internet and the local network. Apache is available for a wide variety of operating systems and is free and open-source.
- **PHP scripting language.** The PHP interpreter generates an HTML web interface file to collect the input parameters (see Fig. 8). PHP is a programming language designed for dynamic web pages. PHP is only executed on the HELAS web server (not on the user's PC).
- **PostgreSQL database technology.** PostgreSQL is used for the authentication and identification of the user sessions (see Fig. 6). PostgreSQL is a free Object-Relational Database Management System (ORDBMS) that is based on Structured Query language (SQL). It offers a real alternative to commercial database systems. For our current and future goals, PostgreSQL would appear to be the best choice.
- **Torque resource manager**. Torque is used to efficiently manage the computational resources. Torque is an open-source software based on OpenPBS.
- **Postfix.** Postfix is used to route and deliver the Torque email notification to the user.
- **Stand-alone application for the MATLAB code.** The MATLAB compiler is used to convert the helioseismology MATLAB codes into self-contained executable applications that are distributed among the MPS compute nodes.

Important features of this system

- Compatible with all user OS platform,
- Multi-user functionality by using secure shell (SSH)
- Easy to use and to fill out Web Graphical User Interface (GUI) (see Fig. 6 and 7)
- The results are saved as .FITS or .MAT files on an MPS disk and can also be viewed as .PNG figures via the Web browser
- Notification to the user by email of the status of the computation (see Fig. 5).

From: saidi@mps.mpg.de Subject: PBS JOB 88.gizon-sxe6.pc.linmpi.mpg.de PBS Job Id: 88.gizon-sxe6.pc.linmpi.mpg.de Job Name: example Begun execution	From: saidi@mps.mpg.de Subject: PBS JOB 88.gizon-sxe6.pc.linmpi.mpg.de PBS Job Id: 88.gizon-sxe6.pc.linmpi.mpg.de Job Name: example Execution terminated

Fig. 5. *Example of a Torque email notification to the user at the end of the job.*

GUI snapshots

Fig. 6. *Authentication Web page.*

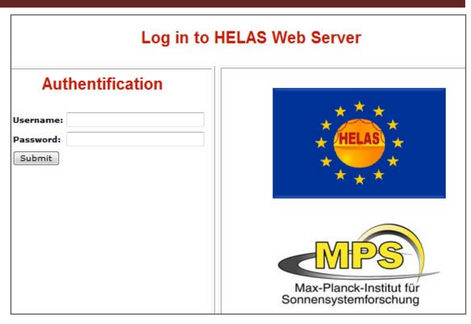


Fig. 7. *List of implemented computations.*

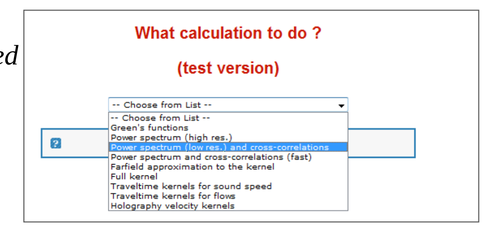
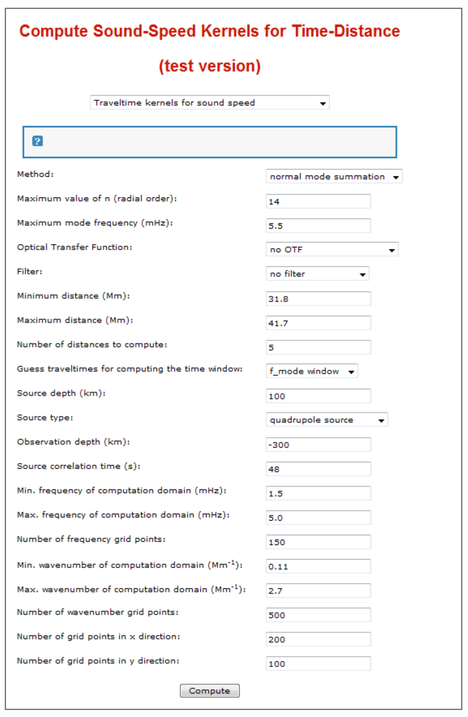


Fig. 8. *Web interface to sound-speed kernel code. Only the most important parameters are requested and the user is presented with meaningful default values.*



Conclusion

We have designed and implemented a concept for running MATLAB codes for local helioseismology over the internet.

Further tests are required to validate the system, especially regarding security issues and user session management. Future work will also focus on a database for archiving and retrieving the computations.

Acknowledgements

This work was funded in part by the European Helio- and Asteroseismology Network (HELAS).

References

A.C. Birch, A. G. Kosovichev and T. L. Duvall, ApJ 608, 580 (2004).

J. Jackiewicz, L. Gizon, A.C. Birch, T.L. Duvall Jr., *ApJ*, accepted (2007).