Scientific Computing in a Web Browser: GALPROP WebRun

Andrey Vladimirov
on behalf of the GALPROP Team
for Colfax International

June 30, 2012

Abstract

As scientific software tools become increasingly complex and computationally demanding, sharing the source code of a scientific project with the community may be insufficient to support peer interest and ensure the appropriate use of the tools. In order to facilitate the use of astrophysical code GALPROP, our group has launched a public online service named GALPROP WebRun. This service, live since August 2010, includes: the ability to configure GALPROP computing tasks in a Web browser; access to a dedicated computing cluster and precompiled binaries for code execution; and user support in the form of online documentation, automated validation tools, and forum/bug reporting online software. This paper reports the details and status of the GALPROP WebRun project as well as our experience with it.

Contents

1 Cosmic rays and the GALPROP code

Cosmic rays (CRs) are energetic charged particles permeating the Galaxy. CRs are related to a large number of outstanding problems in physics and astrophysics. Among these problems are indirect searches for dark matter (DM), charged particle acceleration in astrophysical sources – such as supernova remnants – and the interstellar medium (ISM). The electromagnetic emissions (radio, microwave, X-rays, γ-rays) produced during the propagation of CRs are a significant source of astrophysical information (see Figure 1). At the same time, high energy photons produced by CRs are a source of diffuse background for sources observed by X-ray and γ-ray telescopes. The complex nature of high energy astrophysical phenomena related to CRs requires *reliable and detailed calculations* of CR propagation using a numerical model.

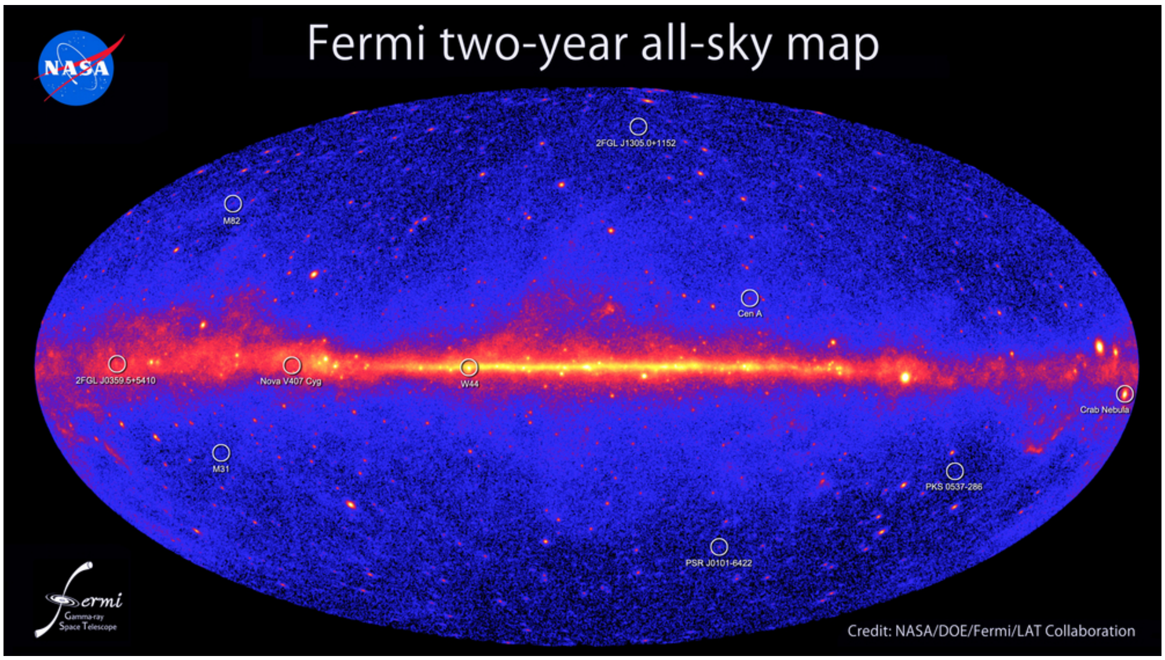

Figure 1: All-sky image in the γ-ray range, constructed from two years of observations by NASA's Fermi Gamma-ray Space Telescope. Point sources (pulsars, active galactic nuclei and other objects) are surrounded by a diffuse component of high energy radiation produced by cosmic ray (CR) interactions in the Galaxy. Modeling the diffuse background requires accurate and detailed CR propagation calculations. GALPROP is an astrophysical code for modeling CR propagation and the associated diffuse γ-ray emission. GALPROP WebRun, discussed in this paper, is a Web service developed by the GALPROP team. WebRun allows to configure GALPROP via a Web browser and run GALPROP calculations on a dedicated computing cluster hosted by the team.

GALPROP is a numerical code for calculating the propagation of relativistic charged particles and the diffuse emissions produced during their propagation. The GALPROP project has been running since the mid-1990s, and since then it has evolved into a sophisticated, efficient, configurable tool for high energy astrophysics. The code incorporates as much realistic astrophysical input as possible together with latest theoretical developments. GALPROP calculates the propagation of CR nuclei, antiprotons, electrons and positrons, and computes diffuse γ-rays and synchrotron emission in the same framework. Each run of the code is governed by a configuration file allowing the user to specify and control many details of the calculation. Thus, each run of the code corresponds to a potentially different "model". The code is written mainly in C++ along with some well-tested routines in Fortran 77.

References to the papers describing the formalism and results of the GALPROP code can be found in Vladimirov et al., 2011 in Computer Physics Communications, Volume 182, Issue 5, pp. 1156–1161. Free arXiv version of this publication is available at http://arxiv.org/pdf/1008.3642.pdf

2 GALPROP WebRun service

2.1 Concept and Purpose

While GALPROP is an open-source project, the installation and utilization of the code requires a significant effort. Dependencies on external libraries such as HEALPix, FITSIO, and other, require careful coordination of library versions. Configuring GALPROP calculations is done via a configuration file with over 100 parameters, and users unfamiliar with the file structure may accidentally produce physically unrealistic configuration and output. In addition, running the code requires a significant amount of CPU time, especially when the sky maps of diffuse γ-ray emissions are calculated.

In order to help experienced as well as novice users of GALPROP to use the code, we have built an Internet-based service GALPROP WebRun. This service enables any member of the scientific community to use the most recent version of GALPROP via a Web browser. Installation of the code locally is not required: the calculations are performed on a dedicated computing cluster employing sixteen 12-core AMD Opteron 6174 CPUs managed by the GALPROP team (cluster specifications are given in Section 2.4). In addition, using the input Web forms of WebRun allows the input parameters to have sanity checks applied via rules written by the GALPROP team, thus minimizing the risk of misconfigured GALPROP runs (see Figure 2 for an example). Each user has a queue that can be used for batch jobs of multiple GALPROP runs, where the user load is dynamically distributed across the available compute nodes. We also provide the latest stable release of GALPROP along with earlier tagged releases to allow cross-checking of results across different versions.

GALPROP WebRun has two purposes. First, it is a learning platform where novice GALPROP users can become acquainted with the scope of GALPROP configuration parameters and dependencies between them. Second, WebRun is a fully functional production system, complete with optimized release version of the GALPROP code and related astrophysical data such as gas maps and interstellar radiation fields. That said, WebRun can produce results appropriate for state-of-the-art research publications. At the same time, advanced users can proceed to download the source code of GALPROP and run it on their own hardware, possibly using WebRun as a graphical interface to code configuration and validation.

WebRun is not intended as the ultimate solution for utilizing GALPROP for astrophysical research, because it does not include any analysis tools and has limited plotting facilities. The results of a GALPROP run (CR distributions and diffuse emission sky maps) are written out in FITS files readable by common astronomical software. Some of these data (e.g., energy spectra of CR isotopes, abundances of CR species, and spectra of diffuse emissions) can be plotted using the WebRun interface. The output of a GALPROP run generally contains much more usable information than given by the built-in plot templates in GALPROP WebRun, so users are encouraged to build their own plotting routines that use the data files produced by GALPROP.

2.2 Internet-based Front End

The front end of GALPROP WebRun operates by communicating with the user via HTML pages with interactive JavaScript-based elements. The pages are generated by a custom PHP-based engine integrated with the phpBB forum software and the Bugzilla bug tracker. Figure 2 illustrates the front WebRun page.

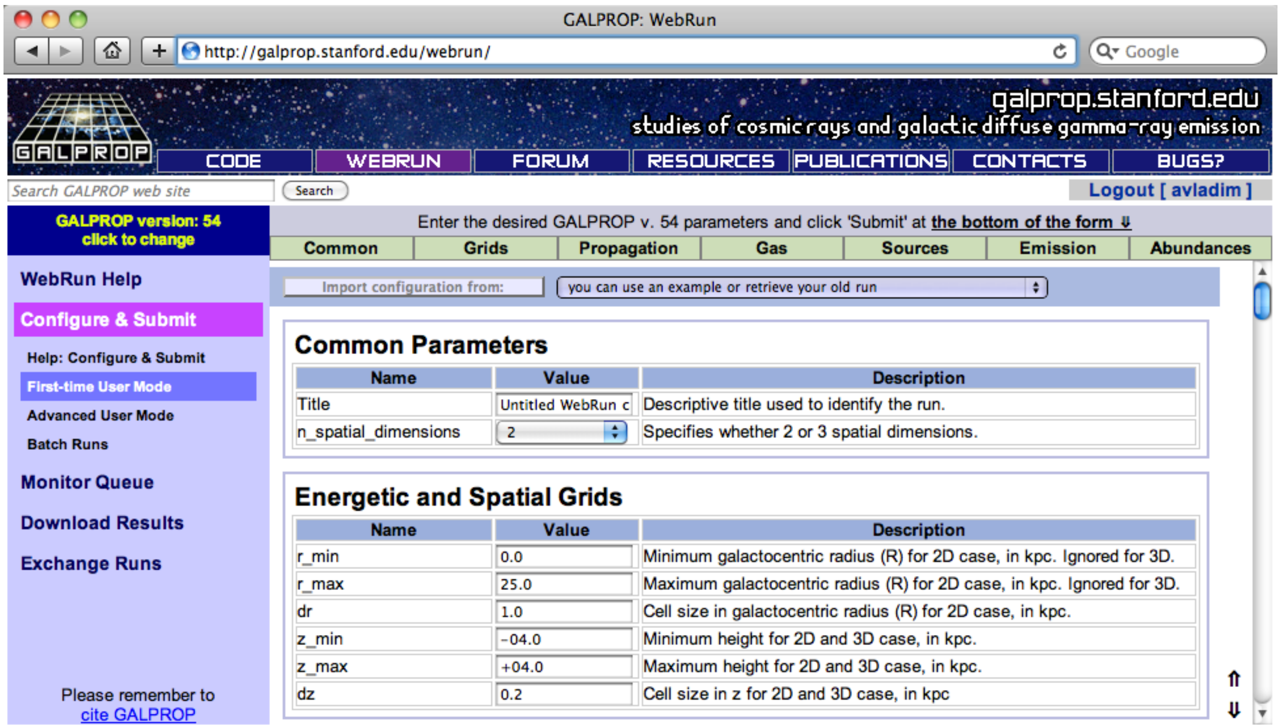

Figure 2: The GALPROP WebRun service allows users to create a GALPROP model using a Web browser, validate the model parameters to reduce the likelihood of misconfigured runs, and execute the GALPROP code on a dedicated computing cluster. The service is accessible via http://galprop.stanford.edu/webrun/.

2.2.1 User authentication

The GALPROP community has been in existence since 2006, and about 150 users were registered prior to the launch of WebRun. Users were registered on the GALPROP forum powered by the popular phpBB forum software. In order to maintain the existing user base, the WebRun user registration and authentication was integrated with phpBB. The authentication in Bugzilla was also tied to the phpBB user base, so that users only need to register once to have access to the forum, bug tracker, and WebRun.

In order to authenticate users, all pages of WebRun include a PHP script, which queries phpBB variables in order to verify whether the user is logged in. This process is illustrated by the code listing below.

```
define('IN_PHPBB', true);
$phpEx = substr(strrchr(__FILE__, '.'), 1);
include($phpbb_root_path . 'common.' . $phpEx);
$user->session_begin();
$auth->acl($user->data);
$user->setup();
if ( ($user->data['user_id'] != ANONYMOUS) && ($user->data['is_bot'] == false) &&
     ($user->data['is_registered'] == true) ) {
  // ... set variables indicating that a registered user is logged in
}
```

2.2.2 GALPROP configuration form

The control panel of WebRun is the GALPROP configuration form shown in Figure 2, which allows users to enter the GALPROP code parameters. The total number of GALPROP parameters well exceeds 100, and therefore a complete and straightforward listing of the full parameter set is impractical.

In order to help users to interact with the configuration form, several interactive features are introduced:

- A 'First-time user mode', where fewer parameters are shown than in the 'Advanced user mode';
- Some parameters become invisible depending on the values of other parameters. For example, when the number of spatial dimensions is set to 2, parameters `x_max` and `y_max` are not used by the code, and they are hidden in the online form; however, the parameter `r_max` is shown;
- Parameters that are fixed and cannot be changed in WebRun are hidden by default. They can be revealed by checking the corresponding check box in the sidebar menu;
- Parameters are grouped by subject to simplify navigation, with group navigation menu at the top of the configuration form. Menu clicks trigger smooth scrolling to the destination group, which helps users to visually track their location on the lengthy form.

In order to render the configuration form and define the parameter visibility and validation rules, server-side PHP scripts parse an XML form specifically designed for GALPROP WebRun. A fragment of the configuration file XML form is listed below for reference.

```
  <gparameter>
                    <name>Title</name>
                    <type>text</type>
                   <brief></brief>
             <description>Descriptive title used to identify the run.</description>
           <default_value>Untitled WebRun calculation</default_value>
    <preferred_way_to_set>firsttime</preferred_way_to_set>
               <visibleif></visibleif>
                   <check></check>
                   <group>common</group>
                <versions></versions>
  </gparameter>

  <gparameter>
                    <name>n_spatial_dimensions</name>
                    <type>integer</type>
                   <brief></brief>
             <description>Specifies whether 2 or 3 spatial dimensions.</description>
           <default_value>2</default_value>
    <preferred_way_to_set>firsttime</preferred_way_to_set>
               <visibleif></visibleif>
                   <check>in:2,3</check>
                   <group>common</group>
                <versions></versions>
  </gparameter>

  <gparameter>
                    <name>r_min</name>
                    <type>real</type>
                   <brief>min r</brief>
             <description>Minimum galactocentric radius (R) for 2D case, in kpc.
Ignored for 3D.</description>
           <default_value>0.0</default_value>
    <preferred_way_to_set>firsttime</preferred_way_to_set>
               <visibleif>$n_spatial_dimensions.eq.2</visibleif>
                   <check>ge:0.0</check>
                   <group>grids</group>
                <versions></versions>
  </gparameter>
```

2.2.3 Running the calculations

In order to submit a calculation to the queue, the user has to press the 'Submit' button at the end of the form. However, before the calculation starts, user-defined parameters are validated on the server side. Validation rules are defined in the above mentioned XML form, and the validation engine is a PHP script interpreting these rules and parsing user-submitted data.

It is important to perform validation on the server rather than the client side, because some of the GALPROP parameters are file names (for, e.g., gas maps), and the ability of the user to manipulate these file names poses a security risk.

Figure 3 illustrates an error message produced by WebRun in response to a misconfigured set of parameters (numerical grid resolution is too low in this case).

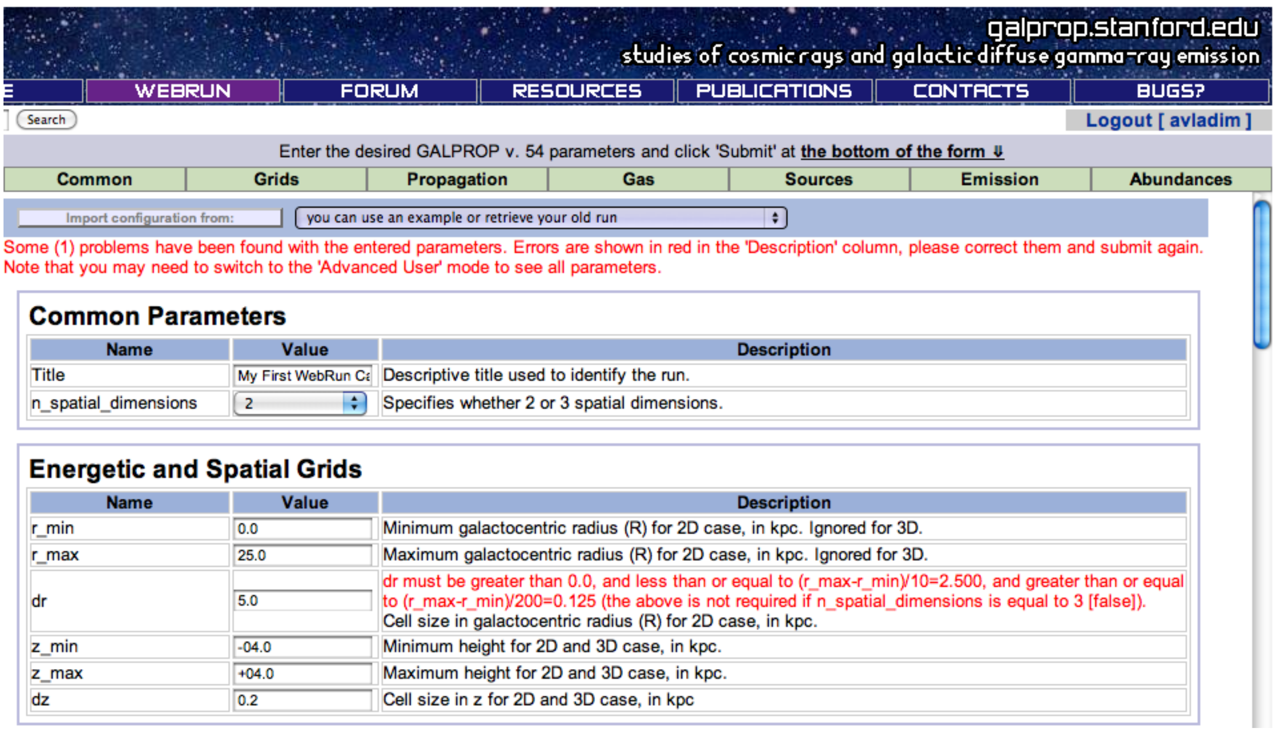

Figure 3: Parameter validation in GALPROP WebRun helps users to avoid misconfigured runs. In this example, the user defined the numerical grid with resolution too low to obtain physically realistic results. WebRun returns the user to the form, pointing out the error. The human-readable error message is auto-generated by the server-side PHP validation script. The script checks parameter validity and generates warnings based on validation rules defined in the XML configuration file.

Once a valid configuration is submitted, the calculation starts on the WebRun cluster located at Stanford University (see Section 2.3). WebRun users can monitor the running output of their calculations and, if necessary, terminate their runs. Figure 4 illustrates the Run Monitor page.

WebRun also supports batch calculations. Instead of using the interactive Web form, the user can submit an archive with configuration files. These files will still be validated. The screenshot in Figure 5 demonstrates the batch upload function.

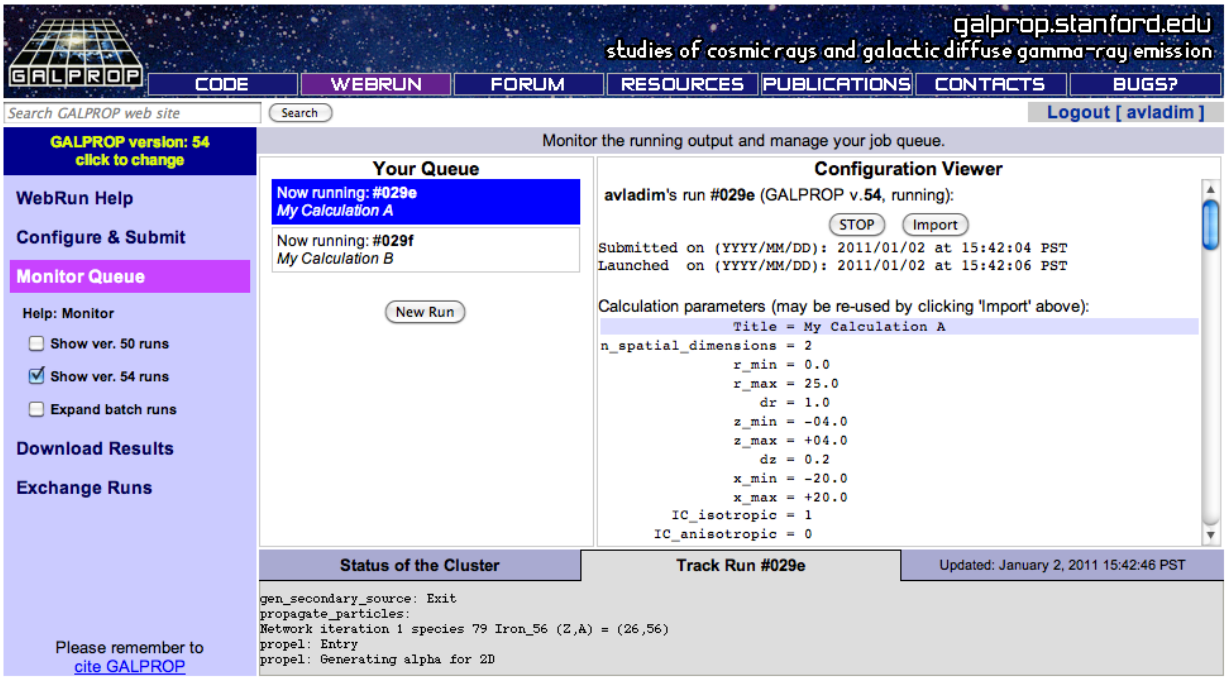

Figure 4: Calculations are executed on the WebRun cluster at Stanford University. Users can monitor the running output of their own runs and terminate them, if necessary.

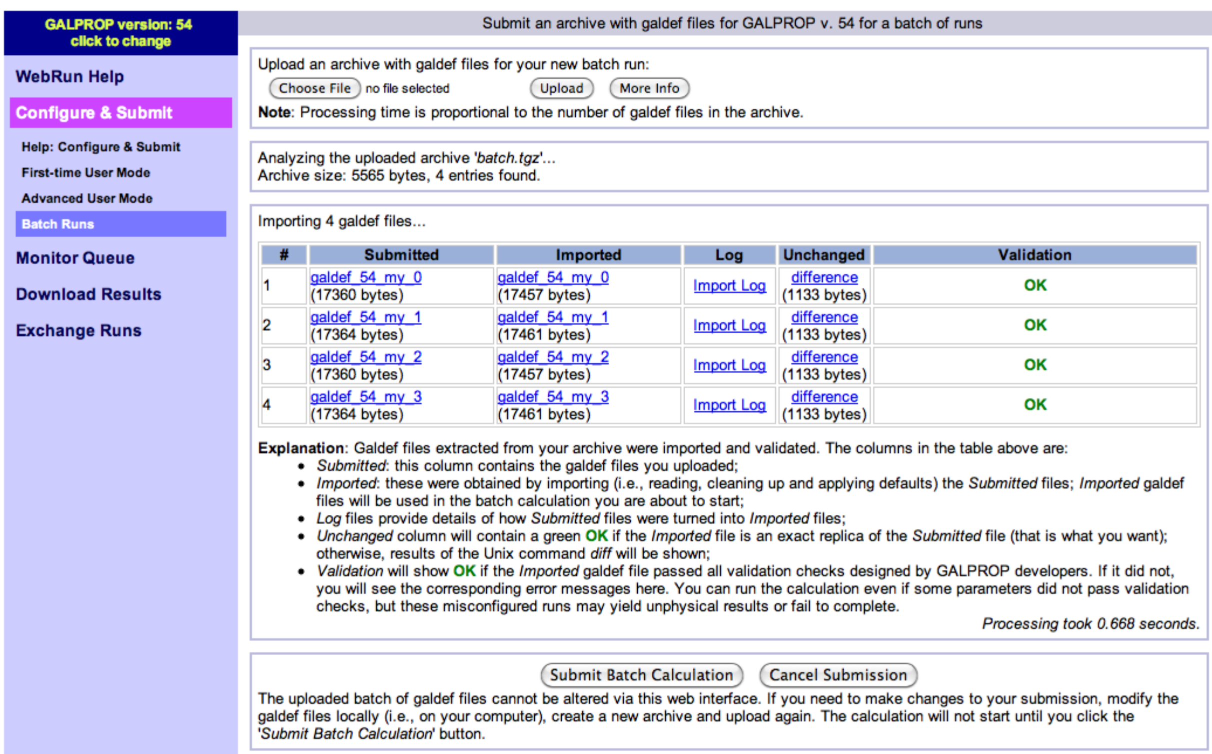

Figure 5: Batch GALPROP calculations can be performed by submitting an archive of configuration files to WebRun.

2.2.4 Viewing and downloading results

GALPROP WebRun has a built-in function for viewing some of the results of GALPROP calculations. Figure 6 illustrates the plot of isotopic abundances, an important output of CR propagation models.

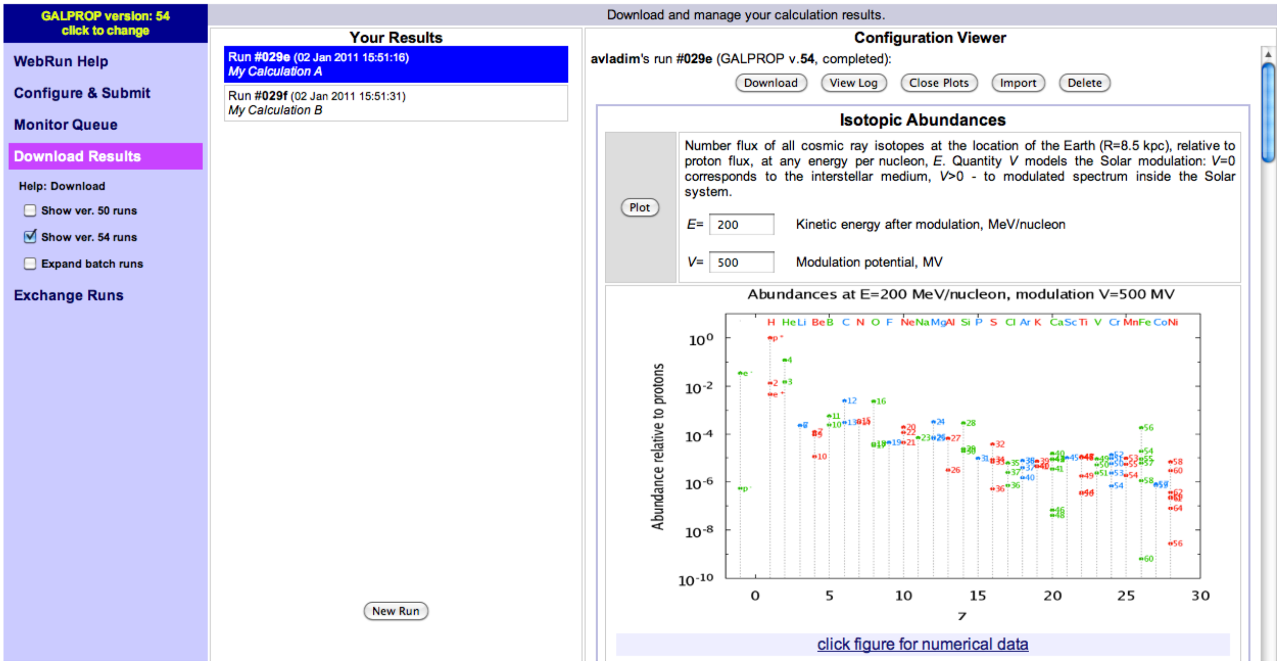

Figure 6: Limited plotting facilities are included in the interface.

However, in order to fully utilize the results produced by GALPROP, the user should download the output files containing the isotopic abundances and γ-ray sky maps in FITS format. The URL of the archive is comprised of the user's ID, WebRun calculation number, and a string of random characters, which prevents unauthorized downloads. At the same time, the URL of the file with results can be shared with anybody, which facilitates collaboration. WebRun also has the 'Exchange Runs' function, which allows run owners to share access to their results with other registered users. Figure 7 illustrates the file download interface.

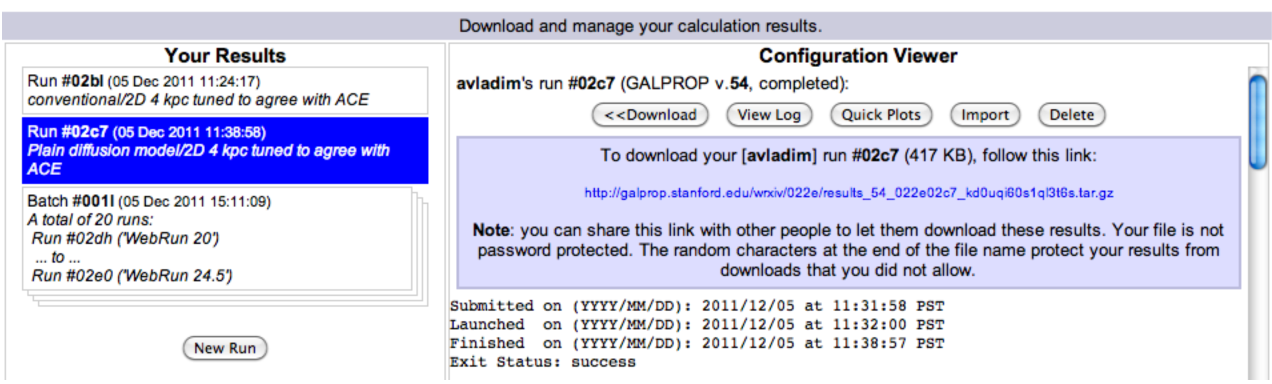

Figure 7: A Tar archive containing FITS files with the results of the calculation can be downloaded from the Web page.

2.3 Back-end Implementation

The server providing the HTTP service for the GALPROP WebRun front end is also running the back-end software responsible for accepting, executing, and serving the results of user-submitted GALPROP jobs. The structure of the WebRun service is shown in Figure 8.

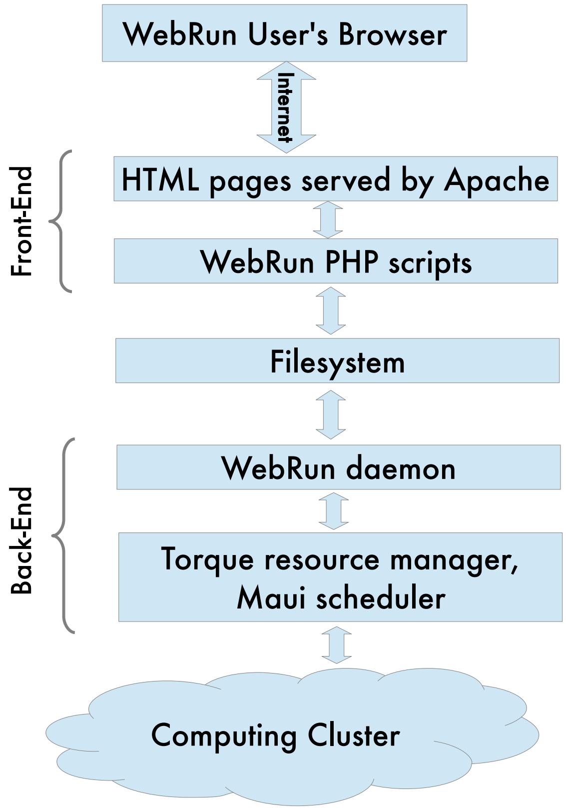

Figure 8

2.3.1 Interaction with front-end software

The interaction between the front-end and back-end software minimal: the back-end daemon only needs to fetch requests for job execution from the interface. For maximum stability and simplicity, this interaction is implemented via the Linux file system. When a job is submitted, the front-end script places a file with job information into a special folder monitored by the WebRun daemon. In order to prevent situations where the file is picked up by the daemon before it is closed by the front-end script, the files are initially created with a temporary file name, and then renamed to the name which triggers the daemon's actions.

2.3.2 Job queue and scheduling

We use the open source Torque resource manager to maintain the queue of calculations. Torque keeps track of jobs submitted to the queue and resources (CPU cores and memory) available on compute nodes. As of today, each WebRun job is allowed to use up to 12 CPU cores and 32 GB of RAM. If enough resources are available on the cluster, and a signal from a scheduler (see below) arrives, Torque submits the next job to the queue.

The scheduler used in the WebRun cluster is the open source Maui scheduler. The use of the scheduler allows three important aspects of cluster work to be controlled:

- When multiple WebRun users have jobs in the queue, the scheduler will try to divide resources equally between these users. This is accomplished by submitting every WebRun job with a unique account string for each registered WebRun user (the argument `-A` of the Torque utility `qsub`).
- When the cluster is used by WebRun as well as for the GALPROP team's internal work, Maui balances resources between the two groups of tasks. This is configured via a Fair Share target for Quality of Service (QoS) in the Maui configuration.
- If the cluster is used only for the group's internal work, sufficient resources are reserved in order to enable immediate launch of WebRun calculations, and vice versa. This is accomplished by setting the value of `MAXPROCS` for the QoS of WebRun and of the team's internal work.

2.4 Hardware Configuration and Operating System

WebRun runs on a cluster comprised of 5 machines:

- The head node, a 2-way machine using 12-core AMD Opteron 6174 processors with 64 GB of physical memory and 16 TB of redundant data storage. The head node is the only machine connected to the Internet.
- Four compute nodes that are connected with private Gigabit ethernet network and equipped with Infiniband links. Each compute node is a 4-way system using 12-core AMD Opteron 6174 processors with 128 GB of physical memory and a modest amount of local storage via redundant disks (2×300 GB mirrored drives) for the operating system and other critical files.

Data sets and other common files are shared via NFS from the head node, as is the common storage area that run results from the compute nodes are written to. Since the current GALPROP parallelization scheme uses a shared memory model (via OpenMP), these AMD-based systems have enabled us to consolidate computational and memory resources within a single unit to enable even high-resolution 3D calculations. In addition, these systems allow us to easily extend the parallel calculations to CPUs and add-in GPU cards using, e.g., OpenCL, or even a hybrid MPI/OpenMP/OpenCL scheme (with some rearchitecting) to use more than one compute node in the future. Similar capabilities were not possible with, e.g., a blade-based system given our power and cost-per-unit requirements. At any rate, this relatively modest system is easily extended dependent on demand for the WebRun service.

The GALPROP WebRun cluster uses the CentOS Linux distribution along OSCAR software suite that is used to install, configure, and synchronize, the software across the cluster.

3 Usage Statistics

GALPROP WebRun has been live for almost 2 years. In the course of this time,

- The user base of the GALPROP project has more than tripled from its original size of 150 members;
- Over 2000 GALPROP calculations were submitted by WebRun users;
- An estimated 5.5 CPU years were used for WebRun calculations;
- Code output in the amount of 300 GB (compressed) has been generated;
- The GALPROP source code and data files were downloaded 150 times;
- 77 bugs and feature requests were submitted to Bugzilla;
- The GALPROP WebRun paper was cited 20 times, including 11 refereed publications.

The plot in Figure 9 illustrates the dynamics of the GALPROP user base. Since the start of the WebRun project, the GALPROP community has been steadily growing by approximately 12 members every month.

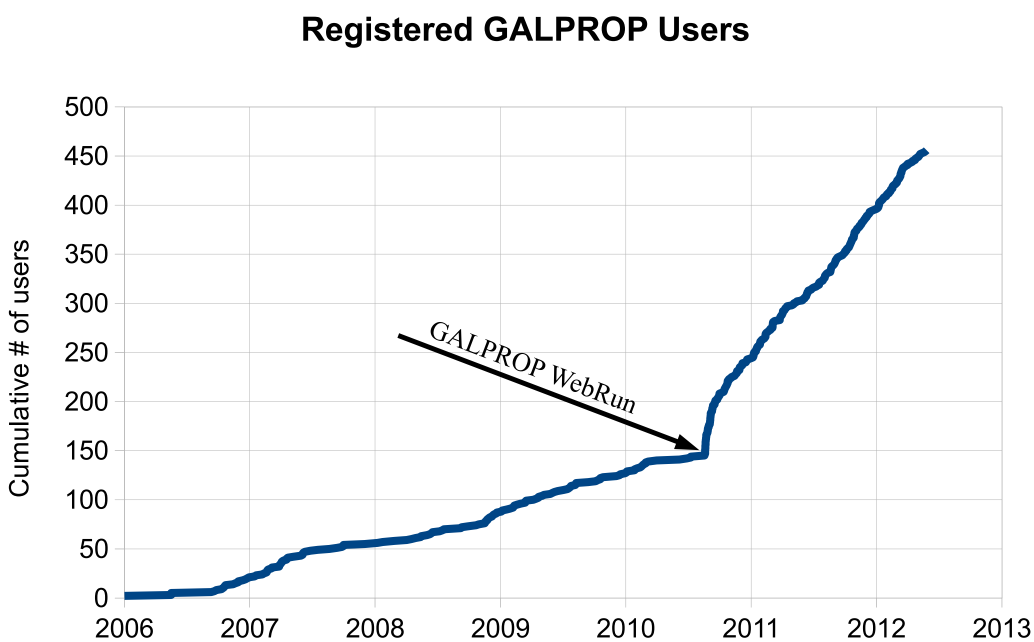

Figure 9: Number of registered GALPROP users since the start of the project. GALPROP WebRun was introduced in August 2010.

4 Support experience

Overall, very little time was required to maintain the GALPROP WebRun cluster in the course of its 2 years of operation. Details of the required support measures are outlined in this section for reference.

Power Supply

Immediately after the installation of the cluster hardware, a stress test revealed that our rack, installed in the basement lab, does not have sufficient electric power supply. The situation was resolved by professional modifications of electric wiring in the room. According to the reading of our UPS units, the power consumed by the 4-node cluster is around 1.5 kW in idle mode and 3.0 kW under full load.

phpBB Forum Support

The user base of WebRun is managed by the phpBB forum software. One issue with this software that we have been experiencing for a long time is 'spam' registrations. Most of these spam registrations were automatic, and installing the Anti-Bot Question mod reduced the amount of spam by at least an order of magnitude. In addition to that, we request that registered users provide their name and institutional affiliation. This helps to eliminate the few human spammers attempting to register at the GALPROP Web site, but forum administrators must manually verify every registration.

As mentioned earlier, the GALPROP user base is managed by phpBB. This software was chosen in order to preserve the user base that existed prior to the launch of WebRun. In order to integrate the phpBB authentication into other GALPROP web site services (WebRun, Bugzilla and code download), some code modification of phpBB and Bugzilla code were necessary. This lead to minor complications during the upgrade of phpBB. These complications could be avoided if an LDAP user base was used. Both phpBB and Bugzilla support LDAP authentication without any modifications.

Operating System Updates

Updating CentOS lead to a problem due to the presence of Oscar packages. The packages for the Torque resource manager, which came with Oscar, conflicted with the Torque packages of CentOS. One of the consequences was the loss of functionality of Torque's utilities, such as `qsub`, `qstat`, etc. The problem was resolved by removing the outdated packages from Oscar. Disk images of compute nodes also had to be manually updated in order to synchronize the Torque client and server version. Since our compute nodes are not connected to the Internet, a local CentOS repository had to be created on the head node and NFS-shared with the compute nodes in order to update them. No other issues with Oscar or the operating systems were encountered.

Hard Drive Failures

In the 2 years of operation, one out of fourteen spinning hard drives of cluster machines has failed. The failure was accurately handled by the hardware RAID1 devices, and no data or service loss occurred.

5 Conclusions

Providing a public Web interface to the GALPROP code and dedicated computing hardware to run the code on was well worth the effort. It resulted in a very positive impact on the adoption, evolution and usage of this astrophysical tool. This leads to the conclusion that Web services providing live access to scientific tools is a relatively unexplored, yet fruitful area of research applications.

Acknowledgements

GALPROP WebRun is a part of the GALPROP project supported by Stanford University and by NASA through grants NNX09AC15G and NNX10AE78G. The GALPROP team includes: Andrew W. Strong and Igor V. Moskalenko, who had developed the first version of the GALPROP code the mid-1990s; Seth W. Digel and Troy A. Porter, who joined the project in the early 2000s; Gulli Jóhannesson, Elena Orlando, and Andrey Vladimirov as more recent additions to the team. Other people have contributed by providing libraries, expertise, and data sets and services over the years: V. S. Barashenkov, Irina V. Malkova, Stepan G. Mashnik, Patrick Nolan, A. Polanski, Olaf Reimer, R. Silberberg, C. H. Tsao, and W. R. Webber. Jeff Wade provides valuable assistance with the system administration for the GALPROP web servers and computing cluster.

The GALPROP team thanks Advanced Micro Devices (http://www.amd.com/) for donated hardware and Colfax International (http://www.colfax-intl.com/) for assistance with the computing cluster serving GALPROP WebRun.

Please visit `http://research.colfaxinternational.com/` to learn more about the Colfax Research project, comment on this article, and subscribe for updates.