

Enhancing SegHidro/BRAMS experience through EELA

José Flávio M. V. Júnior Paulo Ricardo M. Gomes

Laboratório de Sistemas Distribuídos, Universidade Federal de Campina Grande

{zflavio, paulo}@lsd.ufcg.edu.br

Eliane Araújo

Departamento de Sistemas e Computação, Universidade Federal de Campina Grande

eliane@dsc.ufcg.edu.br

Abstract

In this paper, we present SegHidro: a framework for the execution and coupling of hydro-meteorological models on a grid environment, supporting a better management of water resources. Its products and applications will be presented, as well as the computational atmospheric model BRAMS, a key-component of the SegHidro platform. It will be given a brief overview on the portal interface, and described how does BRAMS application works. The link between SegHidro and BRAMS will be explained by showing some structural details of the SegHidro solution. Finally, we shall introduce how do we plan to improve SegHidro/BRAMS experience through EELA Grid Infrastructure as well as the challenges and expectations on the porting of the application.

Introduction to SegHidro

SegHidro (CT-INFO/FINEP) stands for “Segurança Hídrica” (Hydrological Security) and aims at improving the water management of the Brazilian Northeast, a semi-arid region. The project provides a cyber-infrastructure that allows scientists and decision makers to better cope with the water problem of the Brazilian northeastern region, enabling them to better manage the water resources [1].

SegHidro enables, through a grid portal, collaborative work via the coupling of hydro-meteorological computer models, providing access to massive grid-based computer resources using a simple infrastructure in both computation and data [1]. The portal uses OurGrid[1], an open and totally decentralized compute infrastructure, as its grid middleware in order to achieve the necessary computational power to execute the models. For data sharing, SegHidro uses OpenDAP[6], which is a framework that simplifies scientific data accessing.

In alignment with the project goals, openness and decentralization have been fundamental SegHidro values since the beginning of the project. As such, SegHidro is open to the general public, as one can witness in the project portal at <http://portalseghidro.dca.ufcg.edu.br/> [1].

SegHidro Portal

SegHidro provides a framework to simulate a variety of scenarios based on climate predictions and weather forecastings. Through the coupling of computational models, it is possible to improve decisions on water reservoir management, agriculture planning, and flood control [2]. In particular, atmospheric models require a large volume of input data, a great computational power and they produce a big amount of data. The high performance computation that SegHidro needs is currently supplied by OurGrid middleware [1].

The SegHidro Portal is an infrastructure that allows users to execute their applications (computational models) on the grid. The portal interface was built on Apache's JetSpeed [5], that permits the creation of portlets that can easily be integrated and inserted into the portal.

SegHidro Interface

SegHidro Portal is composed of both products and applications. The products can be accessed on the public section of the portal, by just selecting the desired option on the left menu-bar. To access the applications the user needs a password that can be acquired by contacting SegHidro staff by an e-mail provided on the site.

The products available on the portal are weather forecasting and climate prediction datasets, on several formats of data for Brazilian's Northeastern Region, as well as for the states of Paraíba, Pernambuco and Sergipe.

In the application section, five portlets are available: meteorologic portlet, model editing, flow execution, results portlet and job monitoring.

The meteorologic portlet (figure 1) enables a custom run of the atmospheric model BRAMS on the grid. Here, some parameters should be provided by the user, like the spatial coverage and duration of the forecast.

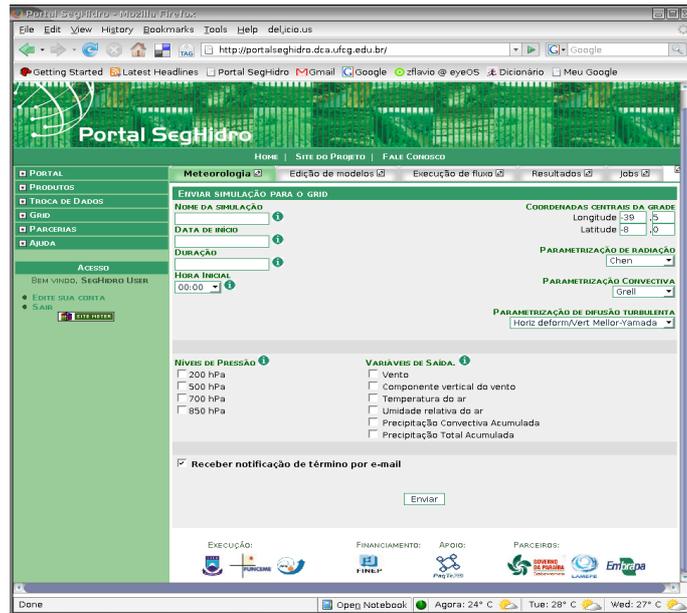


Figure 1: Meteorologic Portlet

The model editing portlet (figure 2) allows the user to upload his own computational models. The user should describe his model with information like the name of the model, its description, executable file, execution command line, input files, output files, etc. The models registered should execute on a non-interactive form, and they can be executed in the execution portlet, described further.

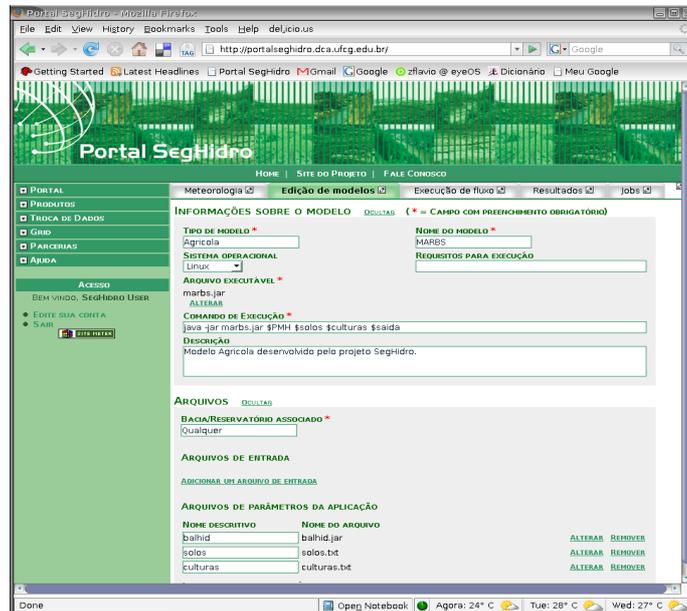


Figure 2: Model Editing Portlet

The flow execution portlet (figure 3) is the interface where the user can build the coupling of computational models (workflow) and send them to the grid. To each input file, an execution flow will be generated. The output of a model can be the input of another model, creating the workflow.

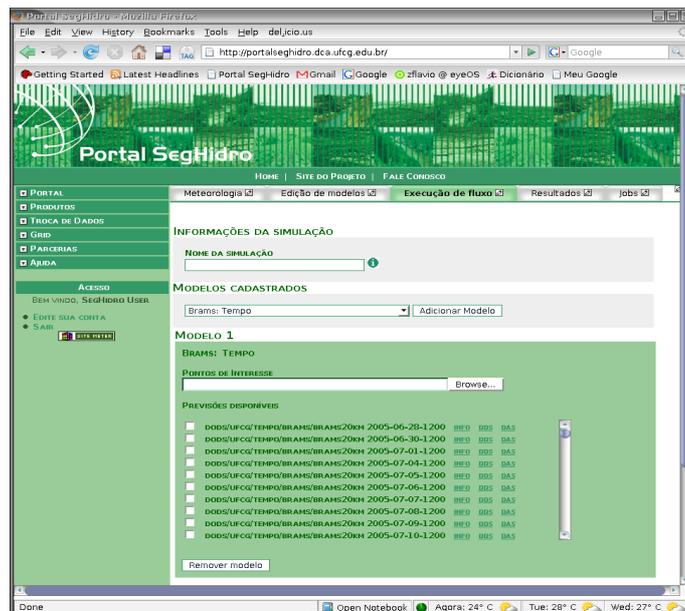


Figure 3: Model Execution Portlet

The results portlet manages the results of the simulations submitted to the grid through the portal. Here are listed all the files resulted from the simulations. They can be downloaded or removed by the user.

Finally, simulations submitted to the grid can be monitored through the job monitoring portlet, that displays details and current status of each job.

OurGrid

OurGrid is an open, free-to-join, cooperative grid in which institutions donate their idle computational resources in exchange for accessing someone else's idle resources when needed. By free-to-join, we mean that anyone can download the OurGrid software from <http://www.ourgrid.org> and join the grid. There is no paperwork or human negotiation regarding what each institution, which we will call site, contributes to and receives from the grid. OurGrid uses the Network of Favors, a peer-to-peer technology that makes it in each site's best interest to collaborate with the system by donating its idle resources. OurGrid can use both interactive desktop computers and dedicated clusters (which may be controlled by a resource manager, such as Maui, PBS, and CRONO [4]).

For now, at least, OurGrid assumes applications to be Bag-of-Tasks (BoT), those parallel applications whose tasks are independent. However, a single OurGrid task may itself be a parallel tightly-coupled application (written in MPI, for example). Although OurGrid does not run parallel tasks across the grid, it may very well run them on a remote site [4].

Users and applications may interact with OurGrid via MyGrid, a personal broker that performs application-level scheduling and provides a set of abstractions that hide the grid heterogeneity from the user [1]. In summary, OurGrid has three main components: the OurGrid peer, the MyGrid broker, and the SWAN security service [4].

OurGrid is in production since December 2004 and now encompasses more than 15 sites that together have around 300 machines. Seven of these sites are SegHidro members. OurGrid current status is available at <http://status.ourgrid.org/> [4].

Introduction to BRAMS

Brazilian Regional Atmospheric Model (BRAMS) is a joint project of ATMET, IME/USP, IAG/USP and CPTEC/INPE, funded by FINEP (Brazilian Funding Agency), aimed to produce a new version of RAMS tailored to the tropics. The main objective is to provide a single model to Brazilian regional weather centers [3].

The BRAMS model simulates atmospheric circulations on limited geographical areas. It has its roots on RAMS (Regional Atmospheric Modeling System). RAMS is a highly versatile numerical model developed by several groups over the years, including the scientists at Colorado State University and ATMET. RAMS is a multipurpose, numerical prediction model designed to simulate atmospheric circulations spanning from large scale down to large eddy simulations (LES) on the planetary boundary layer [3].

Although software portability is central to RAMS and BRAMS, the project targeted PC Clusters under Linux. BRAMS is currently on version 3.2 and the source code is distributed under the CC-GNU General Public License [3].

BRAMS

Compilation and Setup

BRAMS is based in C and Fortran 90. The first step to run the model successfully is to compile the source for the type of machine where the application will be executed. It is strongly recommended that the C and Fortran compilers belong to the same compiler family, due to compatibility reasons.

We performed some compiling tests on some Linux systems and we have got the best results by compiling the package and related libraries with Intel compilers(icc and ifort), free for non-commercial use. We have also compiled BRAMS with GNU compilers(gcc, g95), but we had some runtime problems with it. Another option for compiling BRAMS would be the Portland Group compilers (pgcc, pfg90), but it requires paid license, thus we didn't have the chance to test that.

Before compiling BRAMS, it is necessary to have some libraries installed on the system. It is worth to note that they must be compiled with the same compiler set as BRAMS. The libraries are HDF(used by BRAMS for file handling) and MPI from Argonne (mpich - used for managing parallel execution). They do not require any advanced knowledge to be compiled.

Once BRAMS is compiled, it is ready to run meteorological simulations. For generating accurate output, it should also be provided some updated static input files relative to special conditions, such as sea surface temperature, topography, vegetation, soil moisture, etc. These files are provided by CPTEC and need to be updated on a monthly basis, and the updating can be automatized through shell scripts. CPTEC (Weather Forecasting and Climate Studies Center) is an advanced numeric weather forecasting and high precision climate prediction center, that provides short and medium term forecasting datasets for South America.

The table below describes the file sizes for installing BRAMS:

Description	Size(approx.)
BRAMS and libraries sources	22MB
BRAMS and libraries binaries	40MB
Static input files (compressed)	120MB
Static input files (uncompressed)	600MB
Intel Compilers	170MB

The highlighted terms are the files needed for BRAMS execution, totalizing 640MB.

Execution

The input size (atmospheric files) for a BRAMS simulation will depend on the time range of the atmospheric forecasting simulation. It varies from about 15MB, for a forecasting of 3 hours, to about 200MB, for a forecasting of 6 days. Prior to BRAMS execution, the input files need to be converted to a format that BRAMS understands. This can be done by running an auxiliary script (GRIB2DP) that will perform this conversion in a short time.

BRAMS can be executed in several modes, but the ones we use on our application are the MAKEVFILE mode and the INITIAL mode.

The MAKEVFILE mode takes as input the files that were converted in the previous step and generates the initial state of the atmosphere and boundary conditions over the forecast time for the desired limited area. It does not require though a huge processing, so it can be executed in a plain, single-processor mode, without taking much time.

The INITIAL mode runs the forecast for the desired limited area and time period, using the initial conditions generated in the previous step. It is the step that most consume time and computer processing in the whole sequence, thus it is in this step that MPI is invoked to manage the parallel multiprocessor mode of the execution. The available machines for parallel processing should be accessible from the main node, and the labels of these machines are passed as an argument in the MPI execution line. This step generate analysis files, that are the forecasts themselves. Here the simulation is finished, the next steps are optional for generating human-readable output, and do not take considerable time.

Post processing phase selects desired fields from the Analysis files and produces an output suited for visualization. The post processing tool (RAMSPOST50) produces files for the GrADS visualization tool. After the output is post processed, it can be used in the GrADS for generating images of the forecast.

The following diagram schemes the complete process of the forecasting generation:

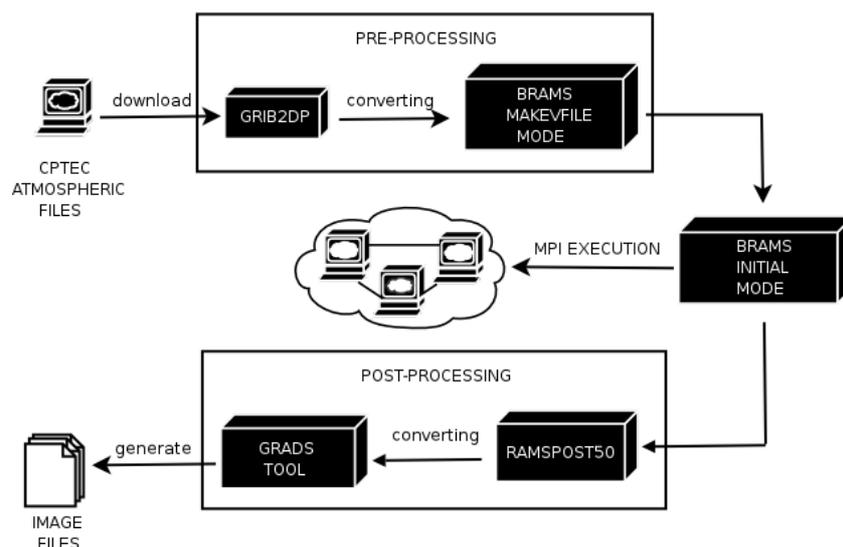


Figure 4: BRAMS execution

All of the steps are currently executed on a grid node, except for the downloading step, that is done automatically by a script that stores the input files on a server in our domains and send them to grid nodes when needed.

The useful results of the processing are images generated by GrADS, as well as RAMSPOST50 processed files. The output varies from about 2MB, for a 3-hour long forecasting, to about 60MB, for a 6-day long forecasting.

SegHidro/BRAMS

We have introduced the details of SegHidro, a project that provides a grid portal that supports researchers and decision-makers on the better management of water resources, and BRAMS, a powerful meteorological application that was specially developed to fit Brazilian regional needs for atmospheric simulations. Now we shall explore the connection between SegHidro and BRAMS, showing that BRAMS is an essential component of the SegHidro solution.

BRAMS Daily Production

As we have mentioned before, one of the products of our portal is the BRAMS daily forecasting for Brazilian's northeastern region (figure 5), as well as for some Brazilian northeastern states, such as Paraíba, Pernambuco and Sergipe. These forecastings covers the desired area's precipitation, temperature and relative air humidity. They are presented in the portal as images, as well as data files on compressed archives.

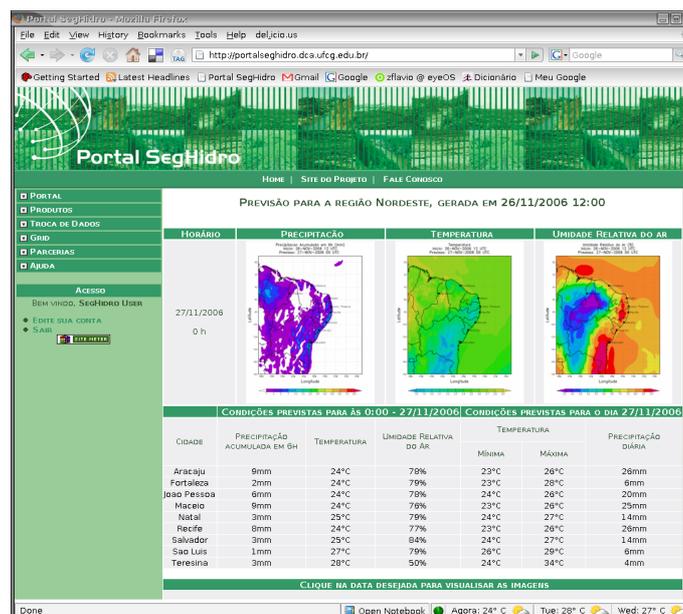


Figure 5: BRAMS daily forecast

Apart from being available on the public section of the portal, for visualization and downloading, the BRAMS daily forecasting is also available as an input on the restricted portlet for the submission of simulations to the grid (execution portlet). This is a really important feature, as a typical hydro-meteorological application is a cascade of computational models that simulates the behavior of the atmosphere, hydrographic basins, aquifers and reservoirs. The start point of the cascade is the atmospheric data, in our case maybe the output of BRAMS.

To produce this data everyday, automatically, in production mode, we administer two HP-Clusters, where we run automatized scripts, evoked by system schedulers, like cron, for managing BRAMS execution on all steps. Each cluster have 10 nodes and double Intel Xeon 2.4GHz processors each node. In one of the clusters, we run a 6-day-long weather forecast, which

takes about 9 hours to get done, and in another cluster we run a 15-day-long weather forecast, that takes about 23hours to get ready. BRAMS is executed just as explained in the section about it, with the INITIAL MODE set to MPI MODE, thus parallelizing the run.

Just after the forecasting is generated, the data is then published on a server(OPeNDAP) and in the system data catalog (Data Locator Service), that are used by the portal to search and retrieve the necessary information.

The Meteorologic Portlet

The meteorologic portlet submits custom BRAMS simulations to the grid. Its simple and friendly interface permits any user to easily run a BRAMS simulation, hiding the details and the complexity that involves the setup of an atmospheric modeling system.

The parameters configured by the user through the meteorologic portlet are validated before sending the simulation to the grid. After verifying if the simulation was already computed, SegHidro Core verifies if there is input data available for the period of time of the simulation. This input data is automatically downloaded from CPTEC, everyday, and stored on a server that is shared via NFS by the grid nodes that are able to run BRAMS simulations.

After validation, the SegHidro Core sends the simulation to OurGrid using its API, that provides the necessary methods to create and manage job execution. The job description contains requirements that makes OurGrid's Broker (MyGrid) request appropriate machines to run BRAMS. These machines have BRAMS previously installed, and share a disk via NFS that contains the necessary input data. The job description also decides whether BRAMS INITIAL MODE will run on parallel mode (MPI), if the selected grid node is a cluster, or if it will run on sequential mode, in case of the selected grid node be an ordinary desktop PC.

Whilst the simulation is running, its status can be monitored on the job monitoring portlet. After the job is finished, the results are presented on the results portlet.

EELA Grid Porting

The grid infrastructure offered by OurGrid is essential to reach SegHidro's goal. The big resource availability combined with OurGrid's ability to run bag-of-tasks applications is a key point on the coupling of hydro-meteorological models, that generally requires huge computational power. The output of these computational models leads decision-makers and researchers to better treat water resources on Brazilian's Northeastern Region.

On the other hand, as parallel (tightly coupled) applications are not supported by OurGrid, the speed performance of the meteorologic portlet will depend on the availability of the clusters on the grid. As most of our community is formed of single-processor machines, and the cluster resources are generally busy running our BRAMS daily production, the simulations on the meteorologic portlet do not always run on their full speed capacity, supplied by the MPI execution. This may lead to a problem if readiness is required as it is the case of weather forecasting. A forecasting is only useful if it is presented before it happens.

It is from that point that comes out the idea of enhancing our service by speeding up Meteorologic Portlet's performance with full support to MPI execution. EELA's grid infrastructure with support to parallel execution appears as an alternative to make SegHidro users more satisfied on the run of custom BRAMS meteorologic simulations.

We plan to accomplish the porting of our Meteorologic application on 3 steps. On the beginning we should get acquainted with EELA's gLite middleware by installing the necessary tools on the computing elements and manually submitting pre-configured BRAMS jobs. After that,

we shall explore gLite's components in order to optimize the run and make the new system compatible with the Portal architecture, by using gLite's resources such as Workload Management System, Data Management System, Data Catalogs, etc. By the time we have a satisfactory, stable and fully working EELA/BRAMS execution system, we are ready to port the changes to the portal. That is when we will make SegHidro's and gLite's API communicate, and finally be able to send meteorologic simulations from the portal to EELA's grid.

Expectations

Besides the various challenges and difficulties that we went through and that we will face, various advantages are being expected on the porting of SegHidro/BRAMS to EELA's grid infrastructure during EELA Grid School.

As we observed before, the portal users will be the most benefited ones, by having the chance of submitting a meteorologic simulations to a worldwide powerful grid infrastructure that will make their experience with the portal much more satisfactory. It will be possible to execute BRAMS faster and attract more users to the portal, spreading the benefits of grid technology to people from other knowledge areas.

SegHidro Team will gain many knowledge on grid technology, by gathering with specialists from all over the world, during the school, that will guide them to produce better quality grid applications.

Relationship between Laboratório de Sistemas Distribuídos(LSD), UFCG and EELA will become narrow, easing the process of integration between gLite and OurGrid, in the future.

References

- [1] E. Araújo, W. Cirne, G. Wagner, N. Oliveira, E. P. Souza, C. O. Galvão, and E. S. Martins, "The seghidro experience: Using the grid to empower a hydro-meteorological scientific network," September 2005.
- [2] W. Voorsluys, E. Araújo, W. Cirne, C. O. Galvão, E. P. Souza, and E. P. Cavalcanti, "Fostering collaboration to better manage water resources," in GCE 2005: Workshop on Grid Computing, September 2005.
- [3] "Brazilian Regional Atmospheric Modeling System". <http://www.cptec.inpe.br/brams/>, 2006.
- [4] W. Cirne, F. Brasileiro, N. Andrade, L. Costa, A. Andrade, R. Novaes, and M. Mowbray, "Labs of the world, unite!!!," Journal of Grid Computing, vol. 4, September 2006.
- [5] "JetSpeed". <http://portals.apache.org/jetspeed-1>, 2006.
- [6] "OpeNDAP". <http://www.opendap.org/>, 2006.