

G-FLUXO: A workflow portal specialized in Computational BioChemistry

E. Gutiérrez^{1,*}, A. Costantini², J. López Cacheiro^{1,*} and A. Rodríguez^{1*}

¹CESGA. Santiago de Compostela. Spain.

²Department of Chemistry, University of Perugia. Perugia, Italy.

Associate Editor: Sandra Gesing and Jano van Hemert

ABSTRACT

The development of a Grid Portal Workflow aware and specialized in Computational BioChemistry is presented.

The P-GRADE Portal (GridSphere based) has been expanded with specific portlet developments adding both support for Computational BioChemistry applications and for different distributed resources is described.

As a first prototype, GROMACS, a versatile package to perform Molecular Dynamics simulations making use of the Newtonian equations of motion for systems with hundreds to millions of particles, has been implemented and supported. Starting from that, specific DAG workflows running GROMACS jobs, demanding very different computational resources on different local and Grid infrastructures (i.e. EGEE, EELA, etc), have been developed and tested. The JMOL based portlet tightly integrated into the portal has been developed in order to help the user in the visualization of both workflow progress and results.

1 INTRODUCTION

In the XX century advances in BioChemistry and Computer Science enabled the development of new mathematical models able to simulate the behavior of complex systems at molecular level, allowing to understand chemical reactions as well as macroscopic properties of such systems. The huge number of atoms and interactions involved in the calculations require an increasing amount of computational resources not always available in local laboratories.

Advances in Computational Science as well as improvements in CPU performance, new architecture designs and the increasing availability of computer power on Grid platforms, together with the adoption of new computational models and algorithms adapted to better exploit parallelism and memory management of such resources, gave a strong contribution to the computing simulation of complex systems.

Nowadays many Graphical User Interfaces for BioChemistry programs have been developed. Special emphasis have been put on the coordinate and collaborative use of different applications in order to perform more complex simulations. Schrödinger Software (<http://www.schrodinger.com>), Scienomics MAPS (<http://www.scienomics.com/Products/maps/index.php>) and

Accelrys Materials Studio (<http://accelrys.com/products/materials-studio>) are commercial examples where an integration of specific scientific applications, execution on remote resources and centralized visualization have been implemented. However, although a lot of work about applications integration (compatibility, input/output format interchange, visualization...) have been done not too much effort have been done with the aim of an easy and efficient execution of applications on the very different computational platforms available today (from a multicore desktop to the heterogeneous and geographically distributed Grid infrastructure passing through the cluster platform located on a lab or on a supercomputer center). In this sense the G-Fluxo project is devoted to the development of a Grid Portal Workflow specialized in Computational BioChemistry where very different computational platforms can be used without the need of very specific computer skills. G-Fluxo is developed trying to use existing and widely used technology in order to avoid very specific requirements for adding computational resources. The P-GRADE Portal (Kacsuk and Sipos (2005)) have been chosen as the starting point of this project so the most widely used Grid middleware is supported. Integration of the cluster platform is based on the SSH protocol and the DRMAA standard (Tröger *et al.* (2007)) operative on most of the clusters running today. Finally a GridSphere based solution (<http://www.gridisphere.org>) such as P-GRADE Portal also lets the development of applications specific portlets fully JSR 168 compliant (<http://www.jcp.org/en/jsr/detail?id=168>).

The first application chosen to be implemented and supported into the G-Fluxo project is GROMACS (van der Spoel *et al.* (2005)). GROMACS is a suite of applications for the simulation of complex systems making use of a wide variety of Molecular Dynamics techniques optimized for different architectures. The variety of simulations needed to solve a particular problem makes GROMACS suitable to exploit different types of computer platforms to solve one single problem. Workflows where each part can be run on very different platforms is possible, requiring to send one part of those jobs to High Performance Computing (HPC) and the other part to High Throughput Computing (HTC).

Computing Grid resources can be considered as a type of HTC platform, on the other hand, local clusters with high speed interconnect networks are examples of HPC platforms. In the present paper a system to split workflows between grid and local clusters in a transparent way for the user is described. Traditionally the management of such workflows is performed manually, increasing notably the effort for the scientist to take

*g-fluxo@listas.cesga.es

advantage of both platforms. The use of the developed web portal provides an abstraction of these systems showing a user friendly, and customizable, interface.

The Web Portal presented in this article makes use of both HPC and HTC infrastructures showing, as an example, an interface for job submission using GROMACS. GROMACS output visualization is done through a specific developed portlet based on Jmol (<http://www.jmol.org>).

The GROMACS and Jmol packages have been implemented also in the COMPCHEM P-GRADE Portal (<http://ui.grid.unipg.it:8080/gridsphere/gridsphere>) web portal in order to test the porting of the portlet applications developed at CESGA.

The paper is organized as follows: in Section 2 the design and implementation of the portal built upon existing technologies is illustrated (a comparison of existing portals is also included); in Section 3 a case study consisting of a workflow based on GROMACS is presented. Our conclusions and future work are summarized in section 4.

2 PORTAL DESIGN AND IMPLEMENTATION

Nowadays the most common way of sending jobs is through the command line, both for grid and local cluster. An alternative to sending jobs through command line is to use a graphical interface. A web portal has been chosen as the graphical interface, because it offers wide compatibility, between different platforms, on the client side. Furthermore each user can access to their data from different places and machines, and they don't need to be worried about new software installation or updates.

In the following subsections, some existing technologies to send simulations through a web portal will be described. Then, with an existing web portal as a base, the modifications in that portal will be described, both to send jobs to local cluster and to visualize simulation job output.

2.1 Existing Technologies

A study of existing technologies has been done where four existing web portals were compared, the results are presented in this Section. Afterwards, the characteristics of the portal chosen as the starting point for our developments will be commented with more detail.

2.1.1 Portal Comparison There are several implementations of web portals for sending jobs to grid. Four Open Source web portals were tested to use as a base for our work: Vine (<http://vinetoolkit.org>), GridPort 4.0.1 (<http://gridport.net/main>), OGCE Release 2 (<http://www.ogce.org>), and P-GRADE Portal. All of these portals have a common execution platform: GridSphere.

As it is shown in the Table 2.1.1 the Vine is remarkable for its interface based on Adobe Flex (<http://www.adobe.com/go/flex>); for example, it has a comfortable file manager for grid. Additionally it has an API for development. With that API it is possible to develop applications for the web portal and also for applications running in a desktop environment. Unfortunately Vine lacks both a standard language and graphical environment for managing workflows. We also tested the GridPort 4.0.1 and OGCE Release 2. Both lack of support for gLite (<http://glite.web.cern.ch/glite>), a middleware widely used in grid platforms (EGEE, ELAA...) and it is currently the main grid middleware used at CESGA.

	OGCE	GridPort	Vine	P-GRADE Portal
Framework	GridSphere 2.1.5	GridSphere 2.0.2	GridSphere 3.1	GridSphere 2.2.10
GLite support	no	no	yes	yes
Workflow editor	yes (through Java Web Start)	no	no	yes (through Java Web Start)
Web interface	portlets	portlets	portlets and Adobe Flex	portlets
VO certificate management	no or difficult	no or difficult	no or difficult	easy (but insecure)

Table 1. Comparison of existing Portal technologies

P-GRADE Portal was the portal chosen as the base for our work because it is Open Source, it supports gLite 3.1, it has a graphical editor for workflows, and it is able to manage the certificates that each user has for each Virtual Organization (VO) of the grid.

2.1.2 Portal Technology Servlets are implementations of Java code that run at the server-side as an answer to a web client request. The result of that execution is a web page. The concept of "portlet" is closely related to servlet concept. The main different is that while a servlet generates a whole web page a portlet generates only one part of a given page. Portlets are defined in the Sun specification JSR 168.

From the user's point of view, a portlet is often represented visually by a window embedded in a web page, with icons to get help, to maximize,... That is called "modes" and can be specified in a XML configuration file. Portlets can be grouped into tabs, this approach allows the creation of separated sections in a web page, providing more versatility in composition, to the administrator of a site and even to a user.

Portlets are usually written in Java and JSP, with some XML configuration files, and deployed, in a portlet container at the server, using a script language as Jakarta Ant.

P-GRADE Portal is implemented using servlets and portlets. In fact, P-GRADE Portal can be considered as a set of portlets which are distributed with its dependencies. Those portlets are mainly focused on grid computing. There are portlets to send jobs and to manage certificates (using MyProxy), files in Store Elements (SE), and user accounts. They are deployed on a container called GridSphere. The version of GridSphere used by P-GRADE Portal is the 2.2.10 version, which it is not the last version and have not good support. GridSphere, in turn, uses the Apache Tomcat (<http://tomcat.apache.org>) as servlet container.

That platform is adequate as a basic working grid environment and to add new portlets which expand the features of the web portal.

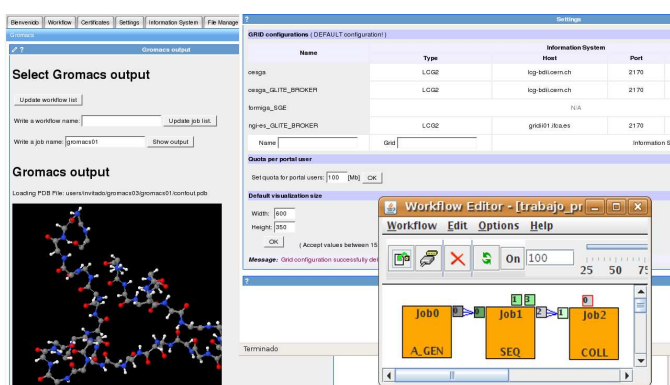


Fig. 1. Workflow application screenshot

There is also a workflow editor, showed in the right-bottom side of figure 1.

A group of tasks connected between them is called workflow. In a workflow, the output of one or more tasks are the input of one or more tasks. In our case, those tasks are mainly jobs which are sent to a grid or local cluster. The workflows can be depicted in a graphical way. This is very useful because it make easy to choose where each job is executed.

It is possible that each job in the workflow needs different resources, for example one needs HTC resources and other need HPC resources, in this case, the job which needs HTC can be run in grid and the other can be run in a local cluster. A user can indicate that in a workflow editor. Also, other job parameters can be changed in a workflow editor, as well as the relations between jobs.

The P-GRADE Portal has a graphical interface for workflows, as shown in the Fig. 1. The workflow is designed in a workflow editor. The workflow editor is implemented in Java, and it run on the client machine. To do that, it uses Java Web Start (<http://java.sun.com/javase/technologies/desktop/javawebstart>) technology.

Each node in the graph, represented by an orange square, corresponds to a job. Each job can be sent to a different grid, being able to indicate explicitly the Computer Element (CE). The workflows implementation is based on a DAG (Directed Acyclic Graph), and uses Condor (<http://www.cs.wisc.edu/condor>) for planning the shipment of those jobs.

This workflow application also facilitates the implementation and execution of parametric studies. In that case you may specify the input parameters of a job to be vary for the study. These parameters, including the intervals between the argument of each parameter vary, and its increase or decrease, are specified in a node in the graph (in one of the orange square that represents one job). For each of these arguments it will generate a job output. To collect all of these outputs there is a "collector". It is also represented by an orange square that is configured as such because it connects the job output (all the exits, with each execution of the job). This method allows to process all the outputs of the parametric study and obtain a single final output result of the workflow.

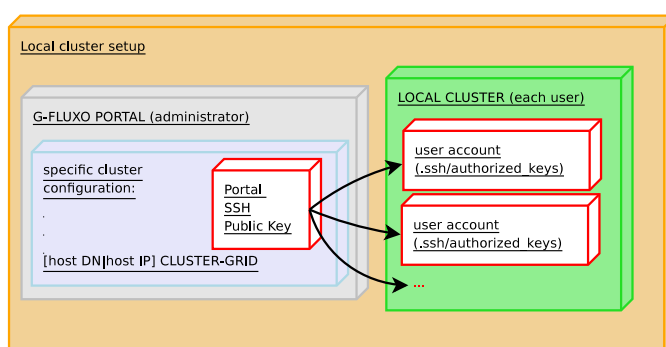


Fig. 2. Local Cluster setup

2.2 Portal Development

Taking the P-GRADE Portal version 2.7 source code as the starting point several add-ons have been developed:

- Integration of the CLUSTER platform
 1. File management and communication through SSH protocol
 2. Distributed Resource Management Systems (DRMS) Job submission and monitoring through DRMAA API based program
- Application specific portlet integration

In the following subsections the way to add a local cluster to the P-GRADE Portal, a new implementation for file management, the local cluster execution via DRMAA, and visualization of job outputs from GROMACS simulations through a Jmol specific developed portlet, will be explained.

2.2.1 Merging Local Clusters and Grid The first step to add a new local cluster is the portal administrator define a new "Grid configuration" in P-GRADE Portal. The only condition in this form is the name having the following syntax:

```
[host_DN|host_IP]_CLUSTER-GRID
```

The second step is the portal administrator has to create a SSH public key for the portal and give that key to each user.

Finally, each user must copy the SSH public key of the portal in the file `.ssh/authorized_keys` of their account at the local cluster. The user should trust in the portal administrator because all their remote accounts are exposed to the portal, but not to the other users of the portal. It is critical that the server where the portal resides must be very secure because it would be able to access to all remote user accounts. This procedure is represented in Fig. 2.

To copy files between local cluster and grid, and between local cluster and server machine ("local" option in that editor), a new syntax has been specifically defined into the workflow editor:

```
cluster:[host_DN|host_IP]:[path_to_file]
```

For each job, the following folder is created in the local cluster:

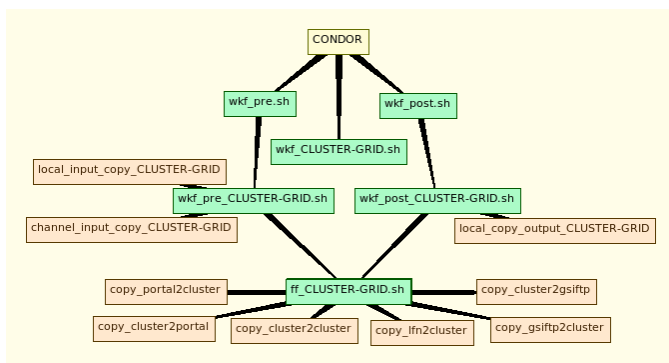


Fig. 3. Scripts and functions diagram implemented for file management and job submission in the local cluster platform

```
[local_cluster]:gfluxo/[workflow_name]
                / [job_name]
```

The files needed for job execution are copied into this folder. So the files specified by the user in job ports have to be sent to this folder by default, or to another path if this is indicated in the workflow job definition.

Each user can indicate, for each job, the necessary resources for the local cluster execution. Those resources are specified by directives in the user script. This directives are parsed through a script configured by the portal administrator. The administrator also has to configure a file to map each portal user to a local cluster user.

One limitation of the workflow editor of P-GRADE Portal is that is not possible to link ports between jobs if one job goes to a local cluster, another goes to the grid and the shared files do not pass through the portal server. In this case a little reference file (semaphore), which is passed through the portal server, has to be used.

For the implementation, the files 'wkf_pre_CLUSTER-GRID.sh', 'wkf_post_CLUSTER-GRID.sh' and 'ff_CLUSTER-GRID.sh' have been created in P-GRADE Portal. A diagram with files and functions modified or created can be seen in the Fig. 3.

The input files are managed by P-GRADE Portal through the 'wkf_pre.sh' script. That script call some functions located in more specific scripts. In the case of local cluster, those functions are 'local_input_copy_CLUSTER-GRID', for unlinked ports, and 'channel_input_copy_CLUSTER-GRID', for linked ports. They are located in the file 'wkf_pre_CLUSTER-GRID.sh'.

The output files are managed by P-GRADE Portal through the 'wkf_post.sh' script. The following function has been introduced in the file 'wkf_post_CLUSTER-GRID.sh' to be called by 'wkf_post.sh': 'local_copy_output_CLUSTER-GRID'.

'ff_CLUSTER-GRID.sh' has been created to separate more specific functions, more dependent on the technology to implement the file management. Those functions are called 'copy_portal2cluster', 'copy_cluster2portal', 'copy_cluster2cluster', 'copy_lfn2cluster', 'copy_gsiftp2cluster' and 'copy_cluster2gsiftp'. Currently, the only implementation is using LCG, for dealing with the Grid platform, and SSH ('scp' and 'sshfs' commands), for local cluster.

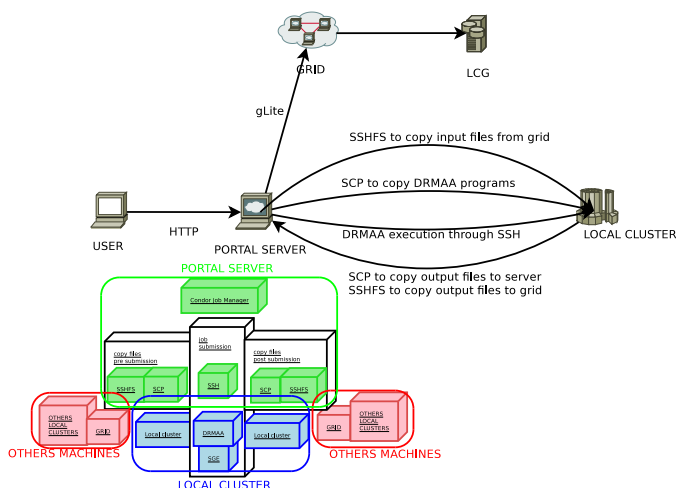


Fig. 4. Platform architecture

Each job is executed on different platforms: Grid or local cluster. The part of P-GRADE Portal to send jobs to Grid has not been altered. The script 'wkf_CLUSTER-GRID.sh' has been created to send jobs to local cluster.

DRMAA is a standard library for the submission and control of jobs to one DRMS. Two programs, coded using DRMAA, were developed: one to send jobs and another to monitor job status. Those programs were compiled, and executed, in the two GridEngine (GE) platforms present at CESGA: SVG (<http://www.cesga.es/content/view/409/42/lang,en>) and Finis Terrae (<http://www.cesga.es/content/view/917/115/lang,en>).

Those two programs are permanently located in the portal. They are sent to local cluster, using the same scripts commented in the previous subsection, when the user job is sent to execute. Then, they are executed by a portal script, 'wkf_CLUSTER-GRID.sh', through SSH. To do that it is necessary to load the remote environment variables needed and then to execute the DRMAA programs.

The user can specify explicitly some requirements for local cluster execution of each job, resources as memory size, CPU time,... Those resources are indicated in the user main script through directives with the following syntax.

```
# gfluxo: [resource_name] [resource_value]
```

Those scripts are executed by the portal through Condor. Condor is integrated in P-GRADE Portal and it is running in the portal server. When a user submits a job, this job is submitted to Condor and the scripts are executed using the files created by the workflow editor and the files upload by the user as input.

An architecture diagram showing the pre/post file management and job submission procedures is presented on Fig. 4. Additionally all the SSH based communication (SCP/SSHFS) channels linked with the specific tasks and the different functions developed in the 'ff_CLUSTER-GRID.sh' file are also shown.

2.2.2 Visualization GridSphere was used as a framework for the development of a specific portlet for the visualization of the GROMACS output. For that, two code files, one in JSP and another

in Java, and the corresponding configuration files were created. Also a Jakarta Ant (<http://ant.apache.org>) script was developed to deploy the portlet in GridSphere. For the specialized visualization needed in the case of GROMACS output Jmol is used. All together is packaged in one tar.gz file for easy distribution.

Jmol is an open-source Java viewer for chemical structures. Inside Jmol the JmolApplet is a web browser applet that can be integrated into web pages and, in our case it has been integrated into a GridSphere Portlet. It supports several input file formats, but does not support the GROMACS format. One popular molecule format is the Protein Data Bank PDB (<http://www.wwpdb.org/docs.html>). The command `pdb2gmx`, of the GROMACS package, is used for the conversion from GROMACS to Protein Data Bank (PDB) file format.

The use of portlets allows us to easily add new tabs and windows associated with simulation applications needed by the user, or applications for file management in a grid, management credentials... Once implemented a portlet for distribution, we create an Jakarta Ant script and everything is packaged in an archive. For installation, the user only has to copy the file, unzip or unpack, and deploy portlets in the container (the GridSphere, in our case) by running this script from Jakarta Ant to be created.

The source code of the Jmol portlet can be downloaded from the G-Fluxo web site (http://gfluxo.cesga.es/download/Gromacs/portlet_jmol_v0.1.tar.gz).

The main class is 'UiJmol'. That class is a child of the 'ActionPortlet' class of GridSphere. The 'UiJmol' class call to 'uiJmol.jsp', where the main layout of the web page is described.

The portlet layout consists of a form, to choose the file to visualize, and of an applet, to render the molecule.

The form was developed using three boxlists (`ListBoxBean` class). The first boxlist is used to choose the workflow, the second to choose the job in that workflow, and the third to choose the molecule file (output of GROMACS). The portlet searches for the files only in the current user account of P-GRADE Portal; it can not access to files in other user's accounts. That Bash script get the name of the user account through the following Java method

```
event.getActionRequest().getRemoteUser()
```

Each list is obtained executing a Bash script. The Bash script is executed from Java code when the user clicks on one of the buttons over each listbox.

To clear the listbox it is needed to use a global variable to execute the sentence `lbb_workflows.clear()` only when the `ListWorkflows` method was already called after the portlet was initiated. Each time a button is pressed, and one element is selected in the previous listbox, the next listboxes are cleared.

The method to visualize created the HTML and JavaScript code necessary to call the Jmol applet with the path of the file. The file has to be copied to a path visible from the web navigator. The path chosen was

```
users/[user_name]/[workflow_name]/[job_name]
/[molecule_file]
```

For the distribution it is needed to unpack the tar.gz file in the 'gridsphere-2.2.10/projects' directory (relative to the root P-GRADE Portal directory). GridSphere directory is deleted by

default in the P-GRADE Portal installation so it is needed to modify the P-GRADE Portal installation script in order to avoid it.

The deployment is done with a Jakarta Ant script. To deploy the portlet, a portal administrator has to execute the following sentences, in the directory 'gridsphere-2.2.10/projects/portlet_jmol_v0.1'.

```
ant install
```

The deployment script is in the file 'build.xml'. That script compiles the Java and JSP source code explained before. Also, it creates JAR files and copies all files to the 'tomcat/webapps/jmol' directory.

The Jmol library is contained in the file 'jmol-11.4.4-full.tar.gz'. That file is unpackaged into '/webapps/gridsphere' directory. This location is needed because it is the root of the P-GRADE Portal, seen by the user's web browser.

3 CASE STUDY

The test application chosen as a case study has been GROMACS as it is stated before. The criteria followed for this election was:

- The application should demand very different platforms (Grid and local cluster) to be run efficiently depending on the input case.
- The application must have a very active development community so it has utilities and visualization tools related with the application that can be used and integrated into the portal.
- it should be desirable to have an Open Source license schema so there will not be any limitation of use.

In order to test the features of the P-GRADE Portal developed under the G-Fluxo project, several Gromacs workflows made up of a chain of three jobs has been run. Each job is run on different platforms (2 clusters, Finis Terrae and SVG, and the EGEE Grid infrastructure (<http://eu-egee.org>)).

These workflows take advantage of all the extra functionalities described previously:

- File Management between different platforms taking into account the dependencies coming from the workflow definition
- Job Submission using gLite middleware and the DRMAA implementation included in the DRMS used in both local clusters (GridEngine, <http://gridengine.sunsource.net>).
- Output visualization including all the intermediate results coming from all the jobs that belong to the workflow.

The developed workflows are based on one of the available GROMACS tutorials (<http://md.chem.rug.nl/education/mdcourse/index.html>). The three jobs are:

1. Vacuum: Energy minimization of the structure.
2. Water: Energy minimization of the solvated system.
3. PR: Relaxation of solvent and hydrogen atom positions: Position restrained Molecular Dynamics.

These jobs must be executed following this order. The workflows used as test case are shown in Fig. 5 and 6.

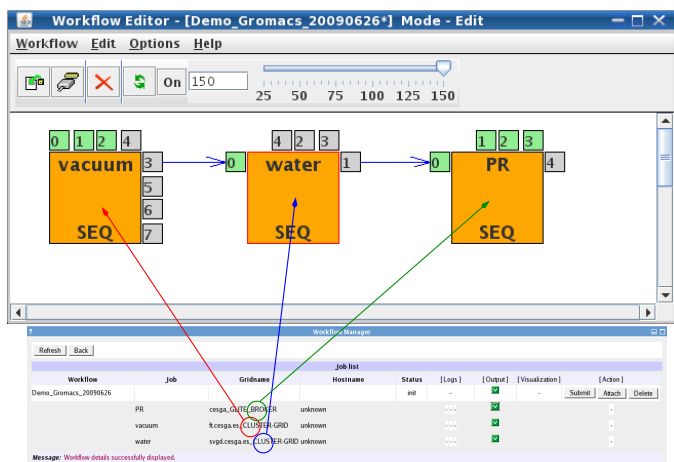


Fig. 5. Workflow definition involving a coordinated execution using Grid (EGEE) and Cluster platforms

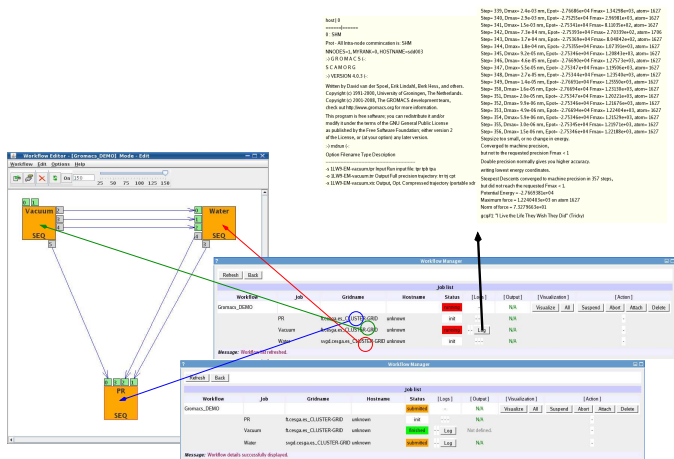


Fig. 6. Workflow definition involving a coordinated execution using only Cluster platforms and visualization of the standard output

In the first case (Fig. 5) the workflow is executed in the Finis Terrae cluster (1st job), the SVGD cluster (2nd job) and the EGEE grid (3rd job). The relations between jobs in this case are made using one link (a semaphore) that defines the dependency job chain. In this case the user must set all the job ports adequately following the syntax described previously in Section 2, taking into account where the job is executed so the portal could be aware of all the information needed for file transfers. This is needed to overcome the limitation present in the P-GRADE Portal file transfer management system that does not allow direct file transfer between different platforms (different Grid middlewares following P-GRADE Portal

nomenclature). Future work that involves a deeper change in P-GRADE Portal will be needed in this sense. In the second case (Fig. 6) only the cluster platform is used although it involves two different DRMS (Finis Terrae and SVG). In this case many link relations are established between the jobs, the output files of the first job became inputs files of the others. In this case (Fig. 7) it is only needed to specify the route of the file (output port) once being automatically assigned to the other job input port.

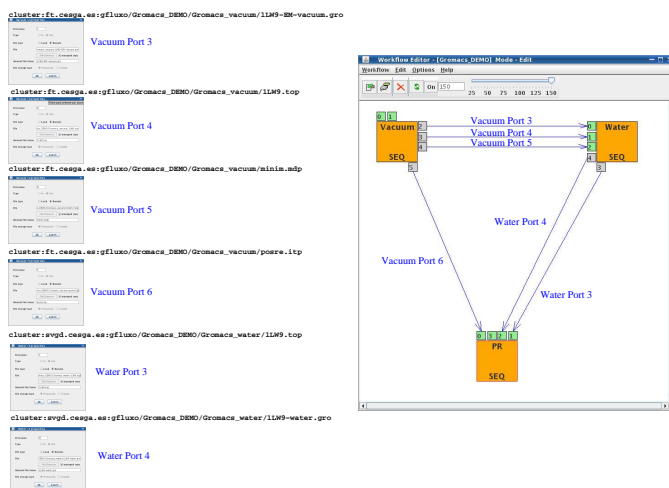


Fig. 7. File Management: Output files from one job became input files of other job. The file transfer need for every job execution is made automatically by the portal.

In Fig. 8 three screenshots coming from the visualization Jmol portlet developed in the GFluxo project are shown. Using this portlet the PDB file carried out from the calculations can be easily visualized and there is no need to wait for the completion of the whole workflow.

All the functionalities available at the JmolApplet are also present in the portlet. Even more, Jmol additional functionalities like the RasMol/Chime scripting language and JavaScript support library can be integrated in the portlet letting the development of a very specific visualization portlet of great help in a specific simulation.

3.1 Integration in COMPChem

In order to test the porting procedure of the GROMACS workflow and JMOL portlet packages developed at CESGA, both packages have been installed and tested in the P-GRADE Portal implemented by COMPChem VO. The COMPChem VO (<http://compchem.unipg.it>) has been created by a group of molecular and material science laboratories committed to adapt their computer codes to run in the EGEE production grid infrastructure. More specifically the goal of designing and implementing grid empowered versions of quantum reactive scattering codes as well as advanced visualization tools devoted to the study of the behavior of complex molecular systems Gervasi and Laganà (2004); Gervasi et al. (2006), is the task of the QDYN and ELAMS working groups of the European Cooperation in

Fig. 8. Workflow definition and output visualization

4 CONCLUSIONS

Web portals to create workflows and send jobs to different Grid infrastructures have been tested. Starting from that a workflow environment to send jobs to grid and local cluster based on P-GRADE Portal has been developed. As a case study in BioChemistry, a GROMACS workflow has been created and tested and a portlet to visualize molecules has been developed and integrated in the portal. The output files carried out from the calculations have been successfully visualized in the new portlet. The GROMACS workflows and JMOL portlet have been also successfully implemented in the COMPCHEM computational environment to test the porting of such applications. Through this paper it has been shown that the combination of a GridSphere based portal like P-GRADE Portal, that facilitates the use of very different computational platforms, jointly with a very specific portlet development (application aware portlet) can be flexible enough to develop specialized portals not only devoted to the Life Science Community. In this scenery, G-Fluxo is able to supply the computational needs that the e-science community could have as:

- Orchestration of the use of many different Computational infrastructures
- Specific support for applications
- Scientific Simulations Modeling: Implementation of different methodologies

Science and Technology (COST) Action D37, called "Grid-Chem" (<http://www.cost.esf.org/index.php?id=189&action=number=D37>).

Thanks to a recent Short Term Scientific Mission (STSM) sponsored by COST D37 Action, the GROMACS package and the JMOL applet have been implemented in the COMPCHEM P-GRADE Portal and made available to the Molecular Science Community supported by the VO. The workflow used as a case study is one the already discussed jobs as shown in Fig. 8:

- Water: Energy minimization of the solvated system.

Thanks to mentioned STSM a set of detailed wiki pages about how to install and run GROMACS in COMPCHEM P-GRADE Portal have been written and made available to the COMPCHEM Community at the following link: [<http://compchem.unipg.it/wiki>].

Two GROMACS tutorials have been written. In the first tutorial the configuration and run procedure for a single job are described and summarized in the following steps:

- Grid configuration, including certificates in MyProxy Server.
- Preparing a job.
- Script to GROMACS on CompChem VO, including how to download the GROMACS application and configure the environments variables.
- Set up of the input and output files of a job.
- Visualization of GROMACS output, using the Jmol portlet.

In the second tutorial a workflow is described. A more complex GROMACS example, with three simulations linked between them, is shown.

A set of video-tutorials, made with the gtk-recordMyDesktop application in Ubuntu 9.04 are also available.

Additional effort is needed in the local cluster platform implementation in P-GRADE Portal. In fact, user access and registration need to be implemented and tested. At present the SSH Public Key Authentication implemented involves some security risks that could be mitigated using a user access and register architecture as the one described in the RETELAB project (Mera *et al.* (2009)).

It is evident that working with workflows implies applications communication. Working on the development of common data formats and conversion routines are critical inside the different life sciences areas. For example tools like Open Babel (Guha *et al.* (2006)) should be integrated on the GFluxo portal. With a conversion tool, job links can be implemented not only as a file copy action but as a format conversion action.

Improvement in workflow support will be also performed and aside from DAG workflow definitions, more complex workflow languages will be supported. A particular effort is focused to link P-GRADE Portal DAG Workflow submission with the Nova Bonita Console (<http://www.bonitasoft.com>), a workflow open source solution that support the standard XPDL process definition language (<http://www.wfmc.org/xpdl.html>).

ACKNOWLEDGMENTS

We would like to thank all the people that have created P-GRADE Portal and released it under an open source license so everyone can contribute to its development. This work is supported by Xunta de Galicia under the project G-Fluxo (07SIN001CT). The grid infrastructures used are those of FORMIGA (07TIC01CT) and EGEE-III (INFSO-RI-222667) projects.

Acknowledgment is also due to the financial support by COST CMST Action D37 GRIDCHEM, through the activities of QDYN and ELAMS working groups, and the EGEE III EU Project under contract IST-2003-508833.

Finally we would like to thank the support from the e-Science Institute for the hosting of the IWPLS'09 workshop.

REFERENCES

- Gervasi, O. and Laganà, A. (2004). Simbex a portal for the a priori simulation of crossed beam experiments. *Future Generation Computer Systems*, **20**, 703–715.
- Gervasi, O., Tasso, S., and Laganà, A. (2006). Immersive molecular virtual reality based on x3d and web services. *Lecture notes in Computer Science*, **3980**, 212–221.
- Guha, R., Howard, M. T., Hutchison, G. R., Murray-Rust, P., Rzepa, H., Steinbeck, C., Wegner, J. K., and Willighagen, E. L. (2006). The Blue Obelisk—Interoperability in Chemical Informatics. *Journal of Chemical Information and Modeling*, **46**, 991–998.
- Kacsuk, P. and Sipos, G. (2005). Multi-Grid, Multi-User Workflows in the P-GRADE Portal. *Journal of Grid Computing*, **3**(3-4), 221–238.
- Mera, D., Cotos, J. M., Varela, J., Coteló, C., and López, J. I. (2009). An integration of several technologies in the architecture denition and deployment of a geospatial grid web portal. In *WORLDCOMP'09 - The 2009 World Congress in Computer Science, Computer Engineering, and Applied Computing*.
- Tröger, P., Rajic, H., Haas, A., and Domagalski, P. (2007). Standardization of an api for distributed resource management systems. In *Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007)*, pages 619–626, Rio de Janeiro, Brazil.
- van der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., and Berendsen, H. J. C. (2005). Gromacs: Fast, flexible and free. *J. Comp. Chem.*, **26**, 1701–1718.