Contribution ID: **6** Type: **Poster**

Science gateways at various levels of granularity for diverse user communities

Monday, 16 September 2013 09:00 (8h 30m)

Description of Work

Molecular Docking simulation programs have significant potential to contribute to a wide area of molecular and biomedical research, including drug design, environmental studies or psychology. AutoDock is one example of a program which allows in silico modelling of intermolecular interactions. Emerging literature shows that AutoDock can be successfully utilized in research strategies for the study of molecular interactions in cancer and for designing drug inhibitors for HIV, for example. AutoDock is a suite of automated docking tools. It is designed to predict how small molecules, such as substrates or drug candidates, bind to a receptor of known 3D structure. AutoDock currently comprises of two discrete generations of software: AutoDock 4 and AutoDock Vina. The latter provides several enhancements over the former, increasing average simulation accuracy whilst also being up to two orders of magnitude faster.

Autodock Vina is particularly useful for virtual screening, whereby a large set of ligands can be compared for docking suitability with a single receptor. In this instance parallelism is achieved by first breaking the set of all ligands into equal sized disjoint subsets. Each compute job then uses a different subset as an input. The ligands in each subset are simulated/docked sequentially on the compute node using the single receptor, whilst a post processing stage can be used to compare the results from all compute jobs.

AutoDock 4 is typically used to accurately model the molecular docking of a single ligand to a single receptor. In this instance the process is composed of 3 discrete stages. First a low complexity sequential pre-processing stage defines a random starting location in 3D space for both the ligand and receptor. This is achieved using a tool within AutoDockTools (ADT) called AutoGrid. The second stage can comprise many parallel jobs, each receiving a copy of the ligand and receptor starting locations which form the input to a genetic algorithm. The algorithm acts to randomly rotate/reposition the ligand and then determine likely docking/binding sites based upon energy levels which are calculated from the original starting locations. This process can be considered a parameter sweep, where the varied input parameter is the initial random rotation of the ligand. Finally, a single low complexity sequential post-processing stage can be used to identify the most likely binding sites by comparing energies from all jobs of the preceding stage.

The aim of the presented work is to develop a science gateway that enables end-user bio-scientist to execute the above described molecular docking simulations from a high level user interface. The WS-PGRADE/gUSE generic science gateway framework has been utilised in this work to provide the baseline technology for rapid development. The presented poster will compare four possible options when developing the customised science gateway.

Option 1 - Deploy the generic WS-PGRADE framework for end-users

In this scenario the end-users can access the generic WS-PGRADE/gUse framework deployed by their system administrator. It is the task of the end-user to design and develop the necessary workflow application to run the docking scenarios. Although the tool provides an intuitive and high level user interface that supports development without needing to deal with low level details (such as certificates, job submission mechanisms etc.), it is well above the expertise of most bio-scientists and requires specific training. On the other hand, if the target community has experienced workflow and application developers then this scenario can be supported by simply installing the framework as it comes “out of the box”.

Option 2 - Pre-create concrete workflows for end-users

This scenario extends Option 1 by pre-developing the required workflows and exporting them to a suitable workflow repository. WS-GRADE offers access to both an internal workflow repository and also to the external SHIWA repository. End-users only need to import these pre-developed workflows to their account, parameterise and execute them. Although it is much simpler than in Option 1, users still need to be familiar with some concepts of the gateway framework, e.g. they do need to understand the workflow concept and should have some awareness of distributed computing infrastructures. This scenario also requires specialised workflow application developers who design, implement and maintain the workflows for the users.

Option 3 - The end user view

WS-PGRADE enables to define templates on top of concrete workflows (by differentiating between fixed and open parameters from the end-user's point of view), and creating applications from these templates. In end-user view the gateway presents these applications as simple web-forms for parameterization by the scientist. This view completely hides the complex details of workflows and DCIs from the end-user. On the other hand, creating an application suitable for end-user view is only a few more clicks when compared to Option 2. As a drawback, the automatically generated forms are relatively rigid, not allowing too much customization. Also, the user still needs to import the workflows from the repository to the individual account. Altogether, the end-user view provides a viable solution for quickly developing customized science gateways without any programming or code development. A publicly available molecular docking gateway utilising this solution has been developed and operated as the AuotoDock portal (https://autodock-portal.sztaki.hu/).

Option 4 - Completely customised gateway developed with the Application Specific Module (ASM) API

WS-PGRADE/gUse has a specific API, called ASM API, that supports the development of completely customized gateways. The ASM API gives access to low level gUSE functionalities for example workflow importing, submission and monitoring. Using the ASM API, customised gateways can be built with reasonable development cost (1-2 weeks development for a custom gateway). This solution was utilized to develop portlets for the University of Westminster Desktop Grid portal, including a set of portlets for molecular docking simulations (https://dg-portal.wmin.ac.uk/liferay-portal-6.1.0/). The customised gateway enables bio-scientists to easily parameterise, submit and monitor workflows, and they can also visualise input and output molecules.

Relevant URL (if any)

https://dg-portal.wmin.ac.uk/liferay-portal-6.1.0/
https://autodock-portal.sztaki.hu/
https://www.sci-bus.eu/

Printable Summary

Science gateways have the potential to offer transparent and user friendly access to a wide variety of distributed computing resources. These tools hide the complexity of the underlying infrastructure from the scientist end-users and let them concentrate on their scientific research problem instead of requiring a steep and sometimes impossible learning curve in complex computing paradigms. Many web and desktop based tools have been developed in the last few years that have been labelled as science gateways. However, close examination of these tools reveal that the level of granularity how end-users can access the applications is rather varied. There are solutions which do not aim to hide the details of the original command line interface and simply provide web based access to the application. On the other extreme, there are custom built portals supporting a single or a small family of applications and providing highly intuitive graphical user interfaces incorporating visualization tools, for example. Science gateways can be developed at various levels of granularity significantly influencing how and by which category of users these tools can be utilised.

Part of the research carried outside inside the European SCI-BUS (Scientific Gateway based User Support) project is to investigate the level of granularity of science gateways that a particular user community requires. Once this level is identified, SCI-BUS supports the development of various end-user gateways in diverse disciplines. This poster will illustrate via the example of a molecular docking gateway "family", how customised science gateways at different levels of granularity regarding their user access can be developed using a generic science gateway framework.

When compared to completely customised "from scratch"development, generic science gateway frameworks provide services readily available, significantly decreasing development time and effort. On the other hand, if these frameworks are extended with customisation methodologies then highly specific gateways can be built on top of them with fraction of the cost. In this gateway development scenario science gateway instance developers select an existing science gateway framework (e.g. the Catania Science Gateway Framework, the HubZero framework, or WS-PGRADE/gUSE) and build customised gateways using low level services offered by the selected framework. These gateways run custom applications of the targeted user community from a suitable high level interface.

A molecular docking gateway supporting several user scenarios (random blind docking and virtual screening) has been implemented on top of the generic WS-PGRADE/gUSE gateway framework. The poster presents and compares four different levels of gateway granularity. Although these implementations are almost identical regarding their performance, significant difference exists regarding the required development effort and also from the perspective of usability.

At the lowest level, the generic framework can be deployed "out of the box"and offered directly for end-users who can build their workflow applications. Although the deployment of the portal is simple and does not require any specific modification or customisation, significant effort and learning curve is needed from the end-users'perspective. In the second scenario, the concrete workflows are developed by specialist workflow developers and published in the WS-PGRADE workflow repository making them accessible for scientists. These workflows can be imported to the end-users'account and parameterised or even customised on demand. The third solution demonstrates the end-user interface of WS-PGRADE where workflow developers can generate custom web-forms on top of the concrete workflows just by dragging and dropping. These web forms hide the complexity of the workflow from the end-user and enable the execution of complex tasks from a high level user interface. Finally, a highly specific set of portlets have been developed using the Application Specific Module (ASM) API of WS P-GRADE incorporating visualisation tools and highly specific user interfaces. In this final solution the generic framework has been fully customised using its high level API to provide an attractive and rich user environment for task execution, monitoring and visualisation. Although gateway development effort was required to provide this customised solution, the utilisation of the generic framework and its API significantly reduced the development time and effort that was required. When compared to developing the same gateway from scratch, relying on the generic gateway framework resulted in a scalable and extendable solution with less than 10% of the effort required.

The presented results are equally relevant and interesting for science gateway framework and instance developers, application developers and end-user scientists. The selected application example demonstrates that granularity is one of the first questions that need to be addressed when developing a science gateway. Selecting the right level of granularity is crucial for providing a usable tool for the targeted community without engaging with unnecessary and time-consuming development efforts. The poster will also demonstrate how a science gateway framework can support gateway instance developers by providing a wide selection of choice and high level programming and customisation tools when defining granularity.

The impact of the presented work is in providing guidelines regarding the granularity of the required gateways, and also in terms of presenting suitable technologies that enable the development of these gateways.

The research leading to these results has received funding from the SCI-BUS project, supported by the European Commission Seventh Framework Programme (FP7) (grant agreement no. RI-283481).

Primary author: KISS, Tamas (University of Westminster, London, UK)

Co-authors: Mr SASVARI, Atttila (University of Westminster); TERSTYANSZKY, Gabor (University of Westminster); Mr BORSODY, Peter (University of Westminster); KACSUK, Peter (MTA SZTAKI); FARKAS, Zoltan (MTA SZTAKI)

Presenter: KISS, Tamas (University of Westminster, London, UK)

Session Classification: Posters display